Project Progress Report

Pediatric Laparoscopic Surgery Simulator

Augmented Reality Laparoscopic Surgery Training

Authors:

Atallah Madi	101156350
Esraa Alaa Aldeen	101151604
Huda Sheikh	101144006
Youssef Megahed	101133061

Group 27

(Biomedical and Electrical Engineering)

Supervised by:

Dr. Carlos Rossa

Report Date:

December 5th, 2023

SYSC 4907

Department of Systems and Computer Engineering

Faculty of Engineering and Design

Carleton University

Table of Contents

Table of Figures

List of Tables

1. Introduction

This progress report provides an update on developing the third version of our capstone project, a state-of-the-art Pediatric Laparoscopic Surgery Simulator. This simulator, which incorporates advanced augmented reality technology and machine learning, aims to improve the training process in pediatric laparoscopic surgery significantly. This report begins with an overview of the project's primary goal, emphasizing the critical need for innovative training tools in this specialized field of surgery. We then offer an in-depth review of the background information crucial to understanding the project. This includes an overview of pediatric laparoscopic surgery, the current standards and requirements for laparoscopic surgical training, and an analysis of recent technological advancements in simulation tools that enhance the skill set of laparoscopic surgeons. A key focus of this report is the link between surgical complications and user performance, particularly how inadequate training and feedback contribute to adverse outcomes. Advanced Pediatric Laparoscopic Surgery Simulator could prevent these complications and guide trainees to improve their skills. Therefore, the Progress Description section details our simulator's innovative features and updates, outlining how it addresses these training gaps and contributes to reducing surgical complications.

2. Background (From the Proposal Report)

Laparoscopic surgery, often called ‘keyhole’ surgery, is a minimally invasive surgical (MIS) technique that differs from traditional ‘open surgery.’ Instead of making a large incision on a patient’s abdomen, laparoscopy involves several smaller, 0.5-1 cm diameter incisions, each called a ‘port’ [1]. A trocar placed inside a hollow sleeve or cannula at each incision or port is slipped through to create an access point for surgery [2]. A trocar is a pen-shaped instrument (triangular at one end) placed inside a hollow tubular instrument [2]. An image of a trocar is shown below in Figure 1.

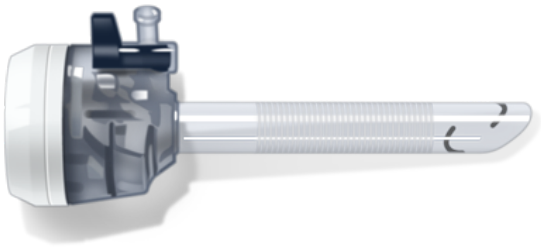


Figure 1 : Image of a trocar.[2]

In these incisions, the laparoscopic tools, a tube to pump gas into the abdominal cavity, and a specialized camera called a laparoscope are placed through [3]. A sample placement of these trocars is shown in Figure 2 below.

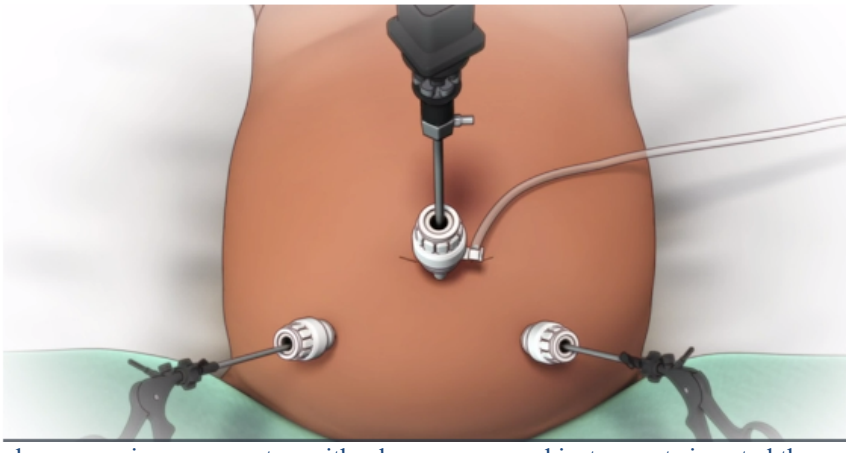


Figure 2: Pediatric laparoscopic surgery setup with a laparoscope and instruments inserted through trocars [3]

Various laparoscopic instruments available today are made of durable stainless-steel material with narrow shafts up to 3-5mm diameter to fit through the ports [2]. Along with a laparoscope, which is a thin telescope equipped with a camera and light source, there are a variety of other tools available for various functions [2]. Needle drivers, composed of handles, joints, and jaws, grasp suturing needles while securing wounds and surgical incisions [2]. Their tips may be curved or straight [2]. Bowel graspers, 3 - 5 mm in diameter, handle and grasp tissue, allowing surgeons to observe and perform various procedures [2]. Other instruments used in laparoscopy include scissors, suturing needles, and surgical meshes [2]. An image of these tools is given in Figure 3 below.

Figure 3: Laparoscopic instruments commonly used today. From left to right, laparoscope (top left), needle grasper (top right), bowel grasper (bottom left), and surgical mesh (bottom right) [2]

Additionally, in laparoscopy, surgeons are limited to four degrees of freedom of motion, which are the yaw, pitch, (counterclockwise/clockwise) roll, and (forward/backward) surge movements [4]. They lose the heave and sway motions due to the pivots that hold the instruments in place [4].

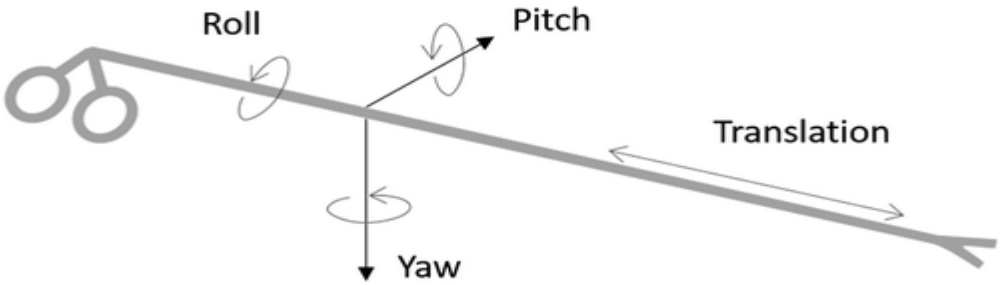


Figure 4:The four degrees of motion surgeons are limited to in laparoscopy are the roll, pitch, yaw, and surge (translation) movements [4]

Regarding postoperative outcomes, due to smaller incisions, laparoscopy is associated with reduced pain post-operation, reduced scarring, reduced hospital stays, and lower incidences of significant wound complications [5]. That means patients can return to everyday life and engage in normal activities much sooner after their operation than open-surgery patients [5].

However, laparoscopy is known to have a steeper learning curve for trainees than conventional open surgery, and each patient-specific procedure can present additional complications for surgeons and trainees [6]. Thus, training for laparoscopy is different from conventional open surgery training in several ways. Trainees must work within restricted spaces and exercise less intuitive and limited movement through the trocars [7]. They must also enhance their hand-eye coordination capabilities due to the added challenges of operating while looking at a screen instead of operating under direct vision, like in open surgery [7]. There is also an increased need to develop manual dexterity skills, like fine motor skills, since the long instruments used in laparoscopy amplify even small movements and errors made by the trainee [7]. Trainees also experience the fulcrum effect, which describes the inversion of movement while operating [7]. For instance, this refers to when the surgeon moves their hand to the right, and the instrument's tooltip is moved to the patient's left on the screen [7].

Additionally, trainees have to grapple with challenges like only using four out of six degrees of motion, the lack of physical sensation or touch from their hands, and the lack of 3-dimensional images like in open surgery [7]. In a clinical setting, laparoscopic training can largely depend on the variety of cases that arrive at the hospital [6]. Due to the increased need for the unique skills required for laparoscopy, a need arose to develop relevant simulators and training programs. Simulators provide trainees with a safe, ethical, and friendly learning environment to

develop their hand-eye coordination abilities, improve their manual dexterity, and soften the learning curve of acquiring laparoscopic skills [6].

When minimally invasive surgery (MIS) was first adapted for infants and children, training surgeons and residents on pediatric laparoscopy presented additional challenges [8]. The use of smaller workspaces, shorter and narrower instruments, smaller margins of error, and the expansion of MIS in children and infants warranted the development of specific learning approaches and simulators for pediatric laparoscopy [8]. This type of procedure demands a more unique set of operating skills than adult laparoscopy. The abdomens of children and infants present a small workspace whereby damage to nearby vessels, the bowel, and other structures can quickly occur and lead to grave consequences [9]. While the instruments are smaller and narrower, surgeons must still maneuver these within smaller anatomical structures and a restricted operating field [9,10]. Due to these limitations, there also exists a lower tolerance of forces to surrounding tissues [10]. Thus, the demand for the unique skill set to perform the procedure has led to the use and development of pediatric laparoscopic simulators. These simulators must mimic the challenging surgical environments in their setup by using smaller instruments, smaller needles, more sensitive force-sensing devices, and more limited surface areas for training tasks [10].

The Fundamentals of Laparoscopic Surgery (FLS) was a program developed by the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) in 2004 to develop a framework that assesses knowledge and skills acquisition for laparoscopic surgery [11]. The program has been widely regarded as a valid and reliable tool and educational curriculum to train and assess surgeons [11]. The Canadian Association of General Surgeons endorses the FLS program as a valuable and comprehensive training and assessment tool to gauge residents' knowledge and skills in basic laparoscopic training. The FLS examination consists of a web-based cognitive examination,

followed by a manual-skills assessment on five fundamental laparoscopic tasks, namely, the peg transfer task, precision cutting task, ligating loop task, suturing with extracorporeal knot-tying task, and suturing with intracorporeal knot-tying task [11]. For now, and for our group's proposed surgical simulator, the focus will be on the peg transfer task and the suturing tasks with extracorporeal knot-tying and intracorporeal knot-tying [11]. More tasks may be added to our simulator if time, resources, and efforts allow.

The most straightforward task, the peg transfer task, involves using two Maryland grasper dissectors [12]. It gets trainees to grasp each ring with their non-dominant hand one by one and transfer it to their dominant hand in mid-air to place it on a peg on the opposite side of the board [12]. The process is repeated until all rings are on the other side of the board [12]. This process is then repeated in reverse, starting with the dominant hand and then transferring the rings to the non-dominant hand [12]. Penalties are applied if the ring falls out of the field of view or is no longer retrievable, but trainees are still expected to continue the procedure should this happen [12]. If a ring is dropped within the field of view, trainees can pick it up with their hands and continue the procedure [12]. The FLS recommends that trainees complete this entire procedure in 300 seconds, with the time starting from the minute the grasper touches the first ring till the time the grasper lets go of the last ring [12].

In the extracorporeal suturing task, trainees are expected to use two needle drivers (or one Maryland grasper and one needle driver) and endoscopic scissors to place a long suture through two marks in a Penrose drain and then perform three single throw knots (extracorporeally) to secure the slit on the drain [12]. Then, once the three throws are secured, both ends of the suture are cut [12]. Penalties are applied for a knot that falls apart, for deviating from the suturing marks on the Penrose drain, and for improper closure of the slit [12]. This procedure is expected to be

completed within 420 seconds, beginning when the first tool is visible through the camera and ending when the cuts are made to either end of the suture [12].

Similarly, in the intracorporeal suturing task, trainees are expected to use two needle drivers (or one Maryland grasper and one needle driver) and endoscopic scissors to place a short suture through two marks in a Penrose drain and then perform three throws of a knot (one double throw, and two single throws) intracorporeally to secure the slit on the drain [12]. Then, once the three throws are secured, both ends of the suture are cut [12]. Penalties are applied for a knot that falls apart, for deviating from the suturing marks on the Penrose drain, and for improper closure of the slit [12]. This procedure is expected to be completed within 600 seconds, beginning when the first tool is visible through the camera and ending when the cuts are made to both ends of the suture [12]. While a scoring rubric from the FLS is not publicly available, plenty of scoring methodologies developed by medical professionals exist online. They can be used to assess the performance of trainees. For instance, many experts have developed independent scoring rubrics for the five basic laparoscopic tasks [13].

To adapt the FLS training and assessment framework for pediatric laparoscopy, smaller and softer rings for the peg transfer task, as well as the 3 mm graspers instead of the 5 mm ones used for adult laparoscopy, should be employed [10]. The size of the Penrose drain used in the intracorporeal, and extracorporeal suturing tasks should be reduced by 6, and smaller needles and 3 mm instruments should be used [10]. Additionally, the surface area of the Velcro securing the drain to the board should be decreased by a factor of 5 to increase the risk of avulsion that can occur due to the application of even minimal force in pediatric laparoscopy [10]. The timing and penalty scores could be the same as that for adults, except additional parameters or penalties may

need to be added to the scoring rubric for pediatric laparoscopies, such as assigning the trainee a score of zero if avulsion of the Penrose drain occurs [10].

2.1. Description of Current Surgical Simulators

The advent of surgical simulators has opened a new avenue in surgical training. It provides surgical residents and trainees with a safe, trainee-friendly learning environment to practice new instruments, procedures, and skills before operating on actual patients [1]. Due to the rapidly changing nature of instruments and laparoscopy's less intuitive techniques/procedures, it is difficult to use the traditional apprentice approach when teaching laparoscopic skills to novices [1]. Thus, many surgical simulators have focused on developing simulation training for laparoscopic surgery [1]. These have included low-tech and high-tech simulators like video-box trainers, organic simulators (on human cadavers and animal models), hybrid trainers, virtual reality (VR) trainers, and augmented reality (AR) simulators [14].

Video-box trainers resemble traditional box trainers in that they provide a box and holes for trocar insertion and various laparoscopic tasks, but also a camera projected on a monitor for the trainee [14]. While they can help practice hand-eye coordination skills and particular laparoscopic tasks, they need more objective expert feedback and focus only on specific tasks/techniques rather than an entire procedure [14]. They require the presence of an expert to assess the performance of a trainee, which is only sometimes available. The FLS uses the McGill Inanimate System for Training and Evaluation of Laparoscopic Skills (MISTELS) system to train novices on the five core laparoscopic tasks as part of its laparoscopic training program [14].

Hybrid trainers, or virtual reality (VR) trainers, integrate video-box simulators with virtual reality to guide a trainee throughout a performance with interactive haptic feedback and feedback metrics without needing an expert surgeon to be present [14]. The user can practice individual

skills like suturing, knot-tying, etc. or go through an entire procedure in a realistic computer-generated environment in more advanced simulators [14]. Objective feedback can be in the form of time taken for a particular procedure or the strength of a knot [14]. Several VR trainers for different subspecialties have become available, including LapSim and Lapmentor simulators for laparoscopy [14]. Sometimes, these simulators may require additional time and effort to set up and maintain as they are not portable. However, they train residents to make sound decisions, preparing them for intense operating environments [14].

Finally, augmented reality (AR) trainers, which overlay computer-generated information into the user's field of view, connect the virtual world to the real world [14]. Some of the approaches covered by AR simulators include overlaying computer graphics on anatomical structures, mapping instrument paths during a procedure, creating a realistic training environment that involves real instruments interacting with actual artifacts in the surgical environment, and providing objective force, motion, and haptic feedback without the need of an expert observing the performance in real-time [14].

The current implementation of our Pediatric Laparoscopic Surgical Simulator project is an augmented reality (AR) simulator, overlaying the surgical environment in the box trainer with computer-generated information to guide the user through a laparoscopic task, like the peg transfer task. Currently, thresholds from sensory information regarding force, velocity, and acceleration of movement are used to assess trainee performance. Our current focus is to enhance the AR simulator by adding more advanced laparoscopic tasks and integrating it with powerful machine learning/data processing algorithms and an interactive user interface to enable it to recognize surgical gestures and classify a user's performance based on accurately collected expert data.

2.2. Machine Learning and Surgical Simulation

Current literature on laparoscopic surgical simulators has focused on applying artificial intelligence (AI) to surgical simulators. AI aims to mimic human intelligence by learning to predict, classify, and learn a particular task to make decisions [15]. AI can enhance surgical simulators today by assessing a trainee's performance, providing more personalized feedback, and enhancing the visualization of the surgical environment itself [15]. Among high costs and long-term maintenance, current VR simulators like LapSim and AR simulators like ProMIS present other limitations, like insufficient metrics/performance criteria to assess trainees [16]. For instance, the ProMIS AR laparoscopic simulator uses "time of performance," "path length," and "smoothness of motion" to assess trainee performance [16].

On the other hand, the LapSim simulator uses "time of performance," "tissue damage," and "path length" to assess trainee performance [16]. "Time of performance" cannot be the most important criterion with which to assess trainees, especially since the primary area of concern is often whether the trainee has used the correct technique and tied a knot properly [16]. Most current simulators do not accurately assess specific surgical techniques, compare a trainee's performance to an expert's performance and provide constructive feedback that the user can understand or implement [17]. Instead, they assume agreed-upon minimum threshold values for completion time, path length, collisions, force, or velocity to assess a user's performance [17].

Machine learning can evaluate performance more accurately and provide more personalized feedback to the user. For instance, machine learning algorithms can be trained to recognize patterns of movement and behaviour in novice and expert performers to provide more detailed feedback to the user. In this way, incorporating machine learning into our simulator will

provide more instructional and informative feedback to the user, increasing the educational value and usefulness of the simulator itself.

There have been several AI-related applications in surgical simulators in recent years [15]. A group at McGill University developed a machine learning algorithm that classified the skill levels of surgeons during a brain tumour resection task and provided detailed feedback [15]. This was done by getting machine learning algorithms to select important performance metrics from an expert-defined list and use these to distinguish between the skill levels of surgeons in a virtual reality simulator [15]. The algorithms could select a novel list of performance metrics and achieve high classification accuracies when classifying trainee performances along training levels [15]. However, the challenges and problems introduced by applying AI and machine learning to surgical simulation training must be addressed. For one, the lack of valid and objective performance metrics for an algorithm to use, and another, the inability of the algorithm to accurately correlate between a trainee's performance on a simulator and in the operating room [15]. In addition, these algorithms may not consider the variety of methods and techniques trainees can use to achieve a particular goal. Potential solutions to these problems could be to divide a procedure into sub-tasks and obtain surgeon-led consensus on evaluation metrics in laparoscopy.

We aim to integrate our current AR pediatric laparoscopic surgical simulator with machine learning algorithms to 1) classify individual gestures or sub-tasks within a surgical procedure (like the peg transfer and suturing tasks) and 2) evaluate a trainee's performance and compare them to an expert's performance. Our conversation with pediatric surgeons Dr. Ahmed Nasr and Dr. Georges Azzie on September 27th, 2023, focused on how the machine learning algorithms that we will develop can assess a trainee's success in completing each sub-task of a procedure by evaluating their smoothness of movement, patterns of motion, and application of force. Each trainee's

performance will be compared against the performances of actual, standard, expert data we have collected. Our methodology regarding our machine learning approach is explained in Section 6, our Methodology section.

3. Project Objective (From the Proposal Report)

The last iteration of the project focused on implementing a functioning box simulator equipped with proper sensors to measure the characteristics of a surgeon's hand movement and force. The simulator was interfaced with augmented reality to provide simple but overly mathematical feedback to the user through warnings and detailed graphs.

The primary goal of this capstone project is to develop a user-friendly training device for pediatric laparoscopic surgery, integrating an embedded system for real-time guidance and feedback. This iteration aims to reduce the reliance on expert surgical mentors, providing trainees with a versatile and safe platform to improve their surgical skills independently while comparing their performance to professional surgeons. Specific, measurable, functional, and non-functional criteria will be employed to measure progress effectively. Functionally, the device's sensors must accurately track intricate hand movements and measure applied forces on the training platform. This multifaceted sensor system will facilitate real-time feedback through an augmented reality interface to enhance the trainee's learning experience. However, the non-functional criteria will cover factors related to the system's back and front end. The backend evaluates data pipelines, signal processing, and machine learning model validation. The front end assesses the system interface's responsiveness and robustness in delivering consistent feedback, including labelled videos highlighting areas for improvement. A significant project objective is to enhance the existing training setup by integrating new hardware components and then building efficient data pipelines to process data from these integrated sensors and feed it to machine learning models. Lastly, the project's final milestone includes presenting this data through augmented reality to convey critical information and compare the trainee's performance to benchmarks supported by machine learning models. This project signifies a commitment to advancing pediatric laparoscopic

surgical training by combining technological innovation and education, creating a transformative tool with significant potential to enhance surgical skills among trainees.

The design of this project will be executed in consultation with our supervisor, Dr. Carlos Rossa, as well as pediatric surgeons from CHEO and the SickKids Hospital in Toronto, Dr. Ahmed Nasr, and Dr. Georges Azzie.

4. Progress Description

4.1. Hardware Improvements

4.1.1. Microcontroller and Load Cells

To enhance the system's overall performance and efficiency, substantial modifications have been applied to the project's electrical components, addressing both hardware and software aspects. The previous Arduino Mega 2560 microcontroller has been substituted with Arduino Nano board, providing comparable speed and data handling capabilities as outlined in Table 1. This substitution not only maintains the system's capabilities but also results in a significant reduction in the project's footprint, thereby creating additional space for possible future upgrades.

Table 1: Comparison between two Arduino boards [18].

Board	Arduino Nano	Arduino Mega 2560
Microcontroller	ATmega328P	ATmega2560
Operating Voltage	5V	5V
Input Voltage	7-12V	7-12V
Digital I/O Pins	14	54
Analog Input Pins	8	16
PWM Output Pins	6	15
UART Serial Ports	1	4
I2C Interface	1	1
SPI Interface	1	1
Flash Memory	32KB	256KB
SRAM	2KB	8KB
EEPROM	1KB	4KB
Clock Speed	16MHz	16MHz
Size	18.5 x 43.5 mm	101.52 x 53.3 mm
Weight	7 grams	37 grams

The existing load cell faces accuracy and precision challenges in measuring forces, particularly influenced by the contact point's distance from the center. As the contact point moves away from the center, the accuracy of the readings diminishes. In discussions with Dr. Ahmed Nasr, and Dr. Georges Azzie, the surgeons emphasized the crucial need for bidirectional force measurement. The current load cell only gauges downward forces, prompting the change to bar strain gauge load cells. These new cells can effectively translate both compression and tension forces, accommodating measurements in both upward and downward directions as shown in Figure 5.

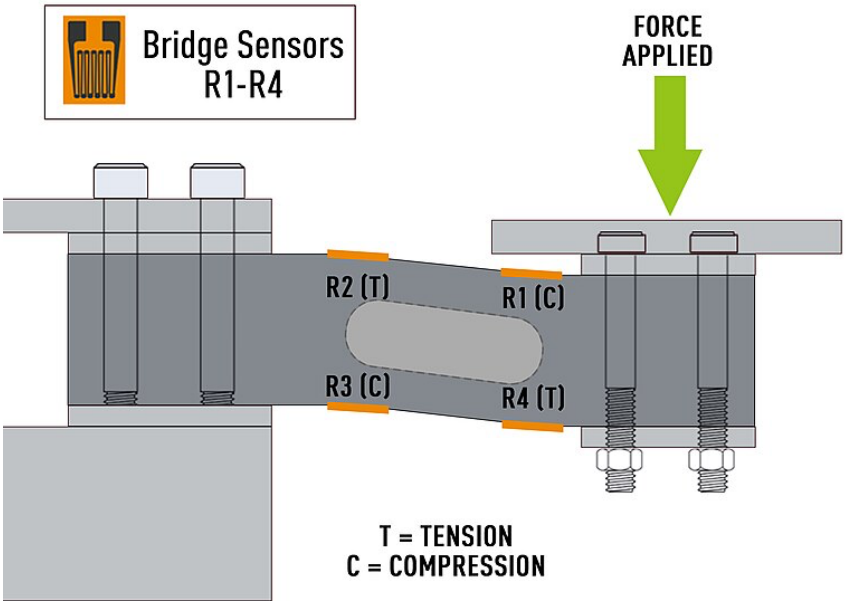


Figure 5: Strain gauge load cell reaction to applied force [19].

In the testing phase, when the contact point is intentionally positioned far from the center, the results indicate a high accuracy level. The new load cells only have a 5% error in such conditions, proving their reliability. The calibration process involved the use of a reference weight which is considered an important step for precise and consistent force measurements. The reference weight serves as a known standard (500 grams) shown in Figure 6, allowing for fine-tuning of the load cells to accurately reflect the applied forces. This approach enhances the

reliability of the system, providing users with confidence in the accuracy of the force readings across diverse operating conditions. Notably, the load cells can provide precise readings to 0.1 gram, highlighting their sensitivity and accuracy. This thorough testing and calibration process addresses concerns raised by the accuracy challenges of the previous load cell design and offering a robust solution for bidirectional force measurement.

Figure 6: Reference weight used in the calibration process [20].

With all of these major improvements to the electrical part of the project, a new schematic has been created to comprehensively capture and illustrate the enhanced system architecture as shown in Appendix A. The updated schematic encapsulates the changes in the microcontroller and load cell amplifiers, showcasing their connections in the revised system configuration. This visual representation serves as a valuable reference for understanding the intricacies of the modified electrical circuitry, facilitating future troubleshooting, maintenance, and potential further enhancements

4.1.2. Computer Aided Design (CAD) Models

The platform redesign focused on two main goals: smoothly adding new load cells and addressing stability concerns. To make room for the new load cells, we strategically added spaces

and mounting points in the design. This ensures that the load cells are perfectly positioned for accurate force measurements, with fixing points both upward and downward to align with any applied force in those directions as shown in Figure 7. Integrating the new load cells also required us to include support structures, like additional fixtures, to secure and stabilize the amplifiers. These fixtures act as strong supports for the amplifiers, preventing any unintended movements or vibrations that could affect the accuracy of force readings. Simultaneously, efforts were taken to address concerns about the platform's stability. Structural improvements, including additional support and bracing, were implemented to ensure the platform remains stable during any task. By carefully designing the placement of load cells and amplifiers, an overall enhancement was achieved to stability and reliability of the platform.

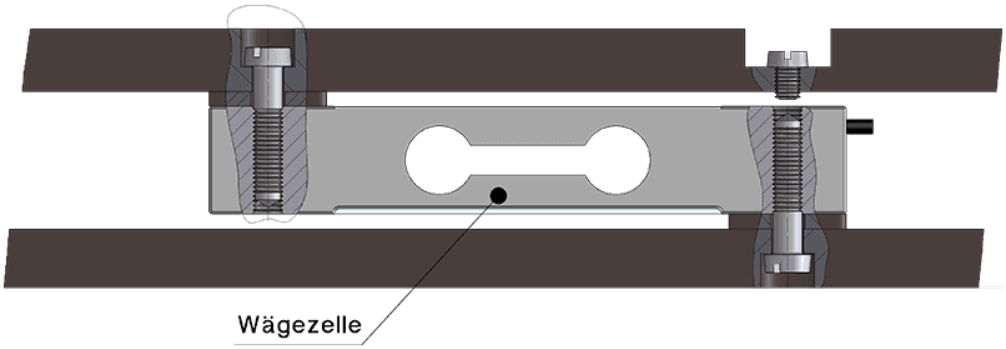


Figure 7: Mounting position for one of the new load cells [21].

Furthermore, due to the changes to the circuit and reduction of the project footprint a new housing was designed to contain the circuit and allocate exit points for the wires going to the sensors located in the trocar points, and the load cells as seen in the project components drawings in Appendix B

4.2. Data Processing Pipeline

4.2.1. Initial Processing

In the latest development of our laparoscopic surgery simulator, the processing of real-time data during simulations is efficiently managed by an Arduino microcontroller. This setup is ideal for seamless data acquisition from each sensor through a signal controller. The system captures a comprehensive dataset, including rotation, position, speed, acceleration, and force parameters. A significant step in this phase is differentiating between each metric's various measures. To handle the raw data collected from the sensors, text files are used for storage. A Python script using the 'tkinter' library was developed to facilitate the rapid processing of this data. This script includes a dialogue window that allows users to select the appropriate text file containing sensor data. Following the selection, the data is cleaned, organized into a DataFrame, and saved as a CSV file using the 'pandas' library, enhancing the efficiency of data visualization and preparation for subsequent algorithms.

4.2.2. Data Segmentation

The subsequent phase in our data processing and preparation pipeline involves a critical step: segmentation. This process entails dividing the entire procedure into three distinct subtasks. Such segmentation proves immensely beneficial for multiple aspects of our system:

A. ***Enhanced Algorithm Feeding:*** By breaking down the procedure into smaller, manageable subtasks, we can feed data into our algorithms in a more structured and effective manner. This segmentation allows the algorithms to process data in chunks more representative of specific aspects of the procedure, leading to more accurate and relevant outcomes.

B. ***Refined User Performance Evaluation:*** Segmenting the procedure enables a more granular evaluation of the user's performance. By analyzing each subtask independently, we better understand the user's strengths and areas needing improvement. This detailed approach ensures that performance assessments are holistic and nuanced, capturing the complexities of the user's interactions with the simulator.

C. ***Targeted Feedback for Improvement:*** One of the primary objectives of our simulator is to provide instructive feedback to users. The feedback can be specifically tailored to each segment by dissecting the procedure into subtasks. This targeted feedback is instrumental in guiding users on improving their performance in each specific area of the procedure, ultimately enhancing their overall skill set.

Segmenting the data into these subtasks was meticulously executed using Python. Here is an overview of how this was achieved:

1. ***Class Initialization - 'DataSegmentation':*** This class forms the foundation for segmenting the data. It initializes with three critical file paths: 'CleanedSensorData_Ref.csv' for the expert's cleaned sensor data, 'user_file_path' for the user to select their procedure data through a file dialogue interface, and 'Labelled Videos.xlsx' for essential labelled information used in segmenting with deep learning methods.
2. ***Loading Data - 'load_data' Method:*** This method is vital for importing necessary datasets into the Python environment for processing. It uses 'pandas,' a powerful data analysis library, to read cleaned reference and user data from CSV files and the labelled information from Excel sheets. The data from two procedure videos, one for an expert

and one for the user, are loaded separately with their respective labels, which is critical for comparative performance analysis.

3. ***Processing Segments - 'process_segments' Method:*** This method takes the loaded data and labels as inputs, creating data segments based on specified time frames in the labels. It iterates through each row in the labels DataFrame, filtering corresponding rows from the data DataFrame. This step is crucial for isolating specific portions of the time series data for focused analysis on defined intervals of each subtask.
4. ***Combining and Saving Segments - 'combine_and_save_segments' Method:*** This method consolidates individual segments into more significant task-specific segments. Using predefined ranges, segments are grouped, forming a comprehensive dataset for each subtask. These concatenated segments are then saved as new CSV files, named to indicate the source video and the specific subtask segment. This organization of data is vital for subsequent independent analysis of each subtask.
5. ***Execution - 'run' Method:*** This method orchestrates the entire segmentation process. It calls the load_data method to import datasets, process_segments to create individual time-based segments and then combine_and_save_segments to compile these into subtask-based datasets. This method processes reference and user data, setting the stage for comparative analysis using the Dynamic Time Warping (DTW) algorithm in later stages.
6. ***Overall Significance:*** The segmentation phase is integral to the project as it structures the data into meaningful and analyzable parts. Breaking down the data into specific segments based on labelled time intervals allows for more precise and focused analysis for each subtask. This structured format is essential for the upcoming steps, where the

DTW algorithm will be applied to each segment for detailed time series analysis and comparison.

In summary, the data segmentation process is a systematic and crucial step. It transforms the cleaned data into well-defined segments, preparing it for the following intricate analysis, including the application of the DTW algorithm and the subsequent evaluation of user performance against the reference data.

4.3. Evaluating Trainee's Performance Using Dynamic Time Warping (DTW)

4.3.1. Time Series Alignment and Performance Comparison in DTW

Data mining involves meticulously sifting through raw data to uncover underlying patterns and extract valuable insights [22]. This process can vary significantly among individuals, leading to notable differences in the duration and pacing of data analysis between users and experts in a laparoscopy surgery setting. Dynamic Time Warping, or DTW, would come into play in this regard. DTW is a method used to calculate the optimal matching between two sequences that may not be perfectly synchronized, often temporal in nature [23]. This technique is prevalent in various fields, including data mining, speech recognition, and financial markets.

DTW leverages dynamic programming to find the best temporal match between elements of a two-time series. An advanced form of DTW, online and dynamic time warping, has been proposed to enhance time series data mining efficiency. This approach involves segmenting a long time series into shorter subsequences and then applying DTW to measure the similarity of each pair of subsequences [24]. To effectively compare a user's performance with an expert, it is essential to align the user's time series with the reference time series, simulating concurrent completion of the procedure. The alignment process is meticulously structured as follows [25]:

A. Each index in the first sequence must correspond to one or more indices in the second sequence, and the reverse must also hold true.

B. The initial index of the first sequence should align with the first index of the second sequence. However, this need not be its exclusive match.

C. Similarly, the final index of the first sequence should align with the last index of the second sequence without being restricted to a singular match.

D. The mapping from indices of the first sequence to those of the second must show a monotonically increasing relationship, and the same applies in reverse. This means if 'j > i' are indices from the first sequence, there must not exist two indices 'l > k' in the second sequence such that index 'i' aligns with 'l' and 'j' aligns with 'k,' and vice versa.

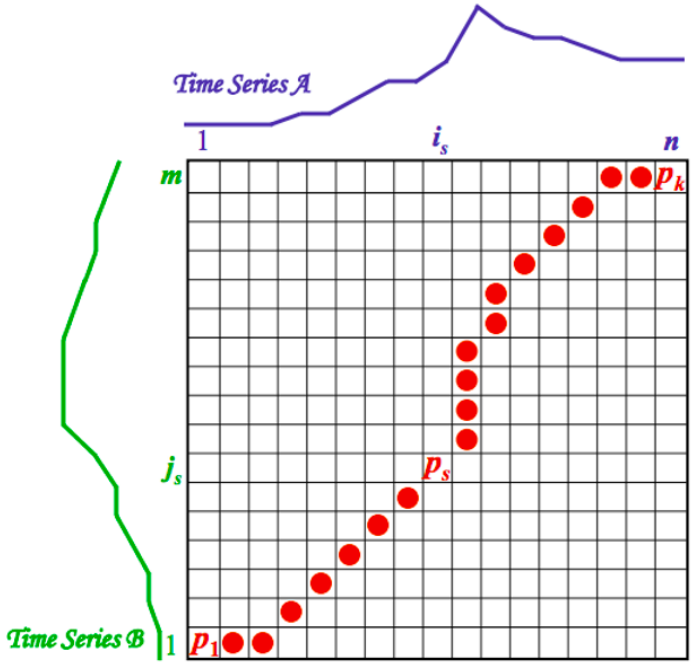


Figure 8: Series A and B distance normalized over time [25].

Mathematically, DTW arranges two sequences, X and Y, into an n-by-m grid. Each point in this grid represents the alignment between elements of X and Y [25]. The warping path W is then computed to map the elements of X and Y to minimize the distance between them. This path is essentially a sequence of grid points.

$$D_{min}(i_k, j_k) = min_{i_{k-1}, j_{k-1}} D_{min}(i_{k-1}, j_{k-1}) + d(i_k, j_k | i_{k-1}, j_{k-1}) \quad (Equation\ 1)[25]$$

Calculating the optimal path to a specific point (i_k , j_k) involves using Euclidean distance. Subsequently, the overall path cost is calculated, representing the cumulative distance over the warping path.

DTW implementation requires consideration of certain constraints to ensure efficiency and accuracy. These include the boundary condition, which ensures the warping path starts and ends with both signals' start and end points, which is shown in the aligned signal in Figure 10; the monotonicity condition, preserving the time order of points; and the continuity condition, which limits the path transitions to adjacent points in time. The warping window condition can also restrict allowable points within a specific width.

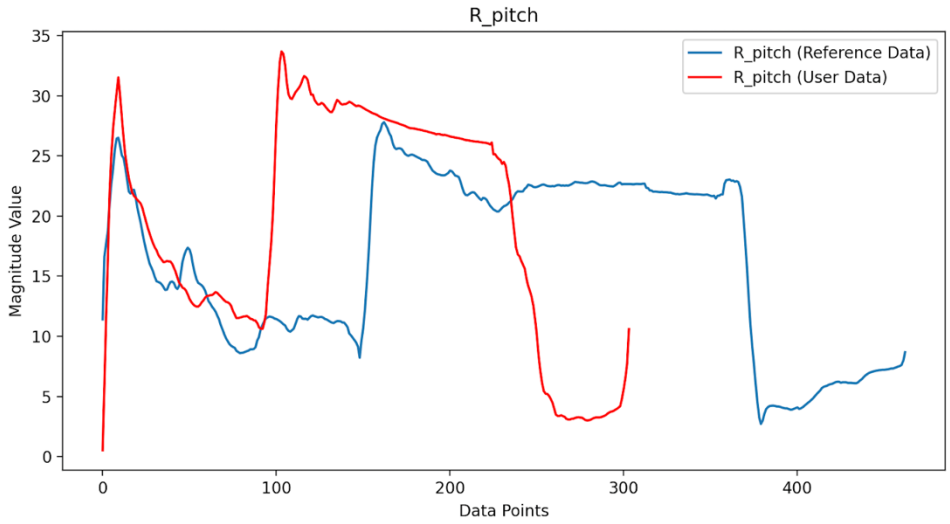


Figure 9: Right laparoscopic surgery grasper pitch movement before applying DTW.

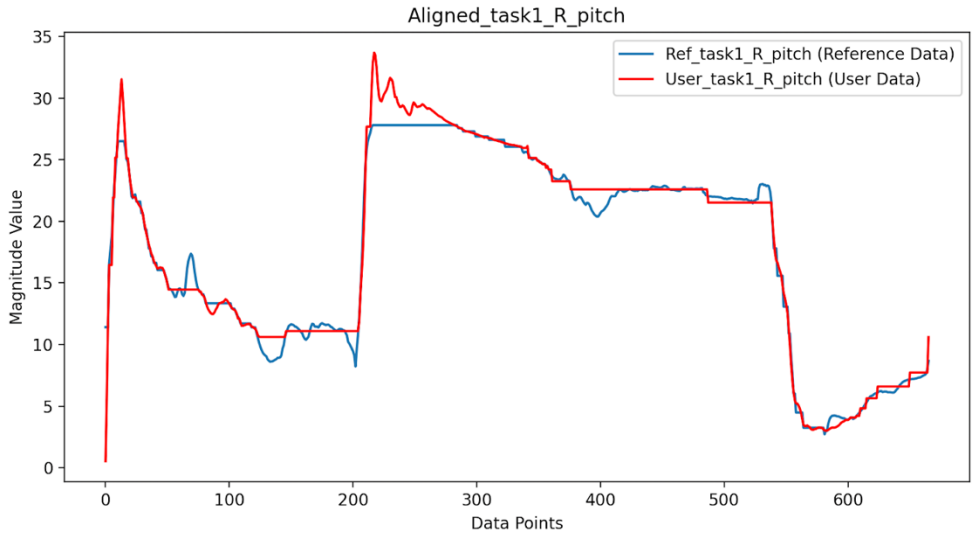


Figure 10: Right laparoscopic surgery grasper pitch movement after alignment using DTW.

In summary, DTW is a powerful technique in data mining for aligning and comparing time series data, especially when there are variations in timing or speed between sequences. Its mathematical foundation and constraints make it a robust and efficient method for analyzing temporal data from the laparoscopy surgery simulator sensors.

4.3.2. TaskPerformanceAnalyzer: Python Class for the Time Series Analysis

A Python Class for Time Series Analysis, the TaskPerformanceAnalyzer class is designed to analyze and compare performance in tasks involving video data and associated time series signals. Upon initialization, the class takes paths to reference and user data in CSV and video formats. It uses the 'pandas' library to load and process the CSV data, dropping unneeded columns and recording signal column names.

The class features a custom method, dtw_distance, to compute the Dynamic Time Warping (DTW) distance between two-time series. This function establishes a matrix and iteratively calculates the minimum distance, reflecting the series's similarity despite potential timing or speed variances.

The align_data method employs the 'fastdtw' function, an efficient DTW implementation for data alignment. This method aligns the reference and user data based on DTW distances and saves the aligned data into a new CSV file. The class also includes a step for normalizing and processing data in fixed-size windows, typically set to 5 seconds. It calculates the DTW distance for each window, categorizing performance and saving results, including statistics like mean, maximum, minimum, and standard deviation of DTW distances, for further analysis.

In the video processing stage, the class uses DTW distance analysis to pinpoint segments in the reference and user videos where performance significantly diverges. This step is done by setting a threshold of the mean DTW distances in all moving windows added to one standard

deviation so that any distance that exceeds this threshold will be detected. Then, these segments are extracted using the ffmpeg_extract_subclip function from the 'moviepy' library. The extracted clips, highlighting weak performance areas, are then combined and annotated with text labels for clarity, as shown in Figure 11 (Youssef is the user, and Atallah is the expert in this case).

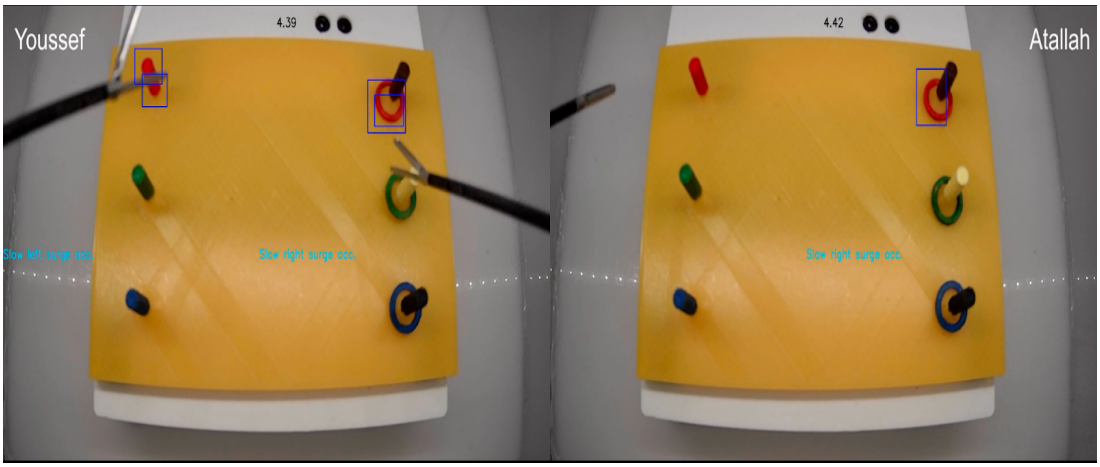


Figure 11: A highlighted clip showcasing areas for improvement in Youssef's performance (the user).

The overall workflow of the class involves comprehensive analysis through aligning time series data, normalizing and segmenting the data for detailed examination, and visually comparing video segments where performance differences are most noticeable. It is beneficial in scenarios where quantitative and qualitative performance assessments are required, as captured on video.

In conclusion, the significance of this section in developing a laparoscopy surgery simulator cannot be overstated. This innovative tool is pivotal in enhancing surgical training by offering detailed, informative feedback on trainees' performance. The simulator facilitates targeted improvement by identifying and displaying clips of areas within each segmented procedural task. This feature is precious as it allows trainees to refine their skills and address potential weaknesses independently, without the constant oversight of an expert surgeon. This autonomous learning approach not only accelerates skill development but also fosters a deeper understanding of surgical techniques, ultimately contributing to the advancement of laparoscopic surgery education.

4.4. User Interface (UI) Design

In the previous implementation of the pediatric laparoscopic simulator, the user interface was designed to be a simple application that relied heavily on the precise location of a user's mouse movements to progress through the program. In addition, the absence of buttons and relevant graphics made for a less sophisticated design that would not work with our requirements or our additions to the data/signal processing element of our program. Thus, this prompted the creation of a new user interface that would use some functionality of the old program but would be based on more sophisticated GUI capabilities from Python's extensive Tkinter library. The subsequent sections below will present the new and improved user interface, along with the features that were added to enhance the functionality of the existing program.

For simplicity, and ease of use and integration, Python was chosen as the selected language to implement the new user interface. Since the old program was built in Python, and the data processing, signal processing, and machine learning steps are also executed in Python, it was decided that the language's tkinter library has extensive GUI capabilities, and coupled with its ttkthemes library, will allow us to develop a powerful user interface. The object-oriented programming paradigm will separate large portions of code written using the functional programming paradigm into distinctly defined classes.

4.4.1. The Design Process of the UI

4.4.1.1. Creating a Prototype

Before initiating the programming process, a prototype for the user interface was developed to develop some sense of organization. However, the actual program looks somewhat different from what we intended, as a result of multiple modifications and design improvements. Figure 12 shows some examples from the prototype we developed using Microsoft PowerPoint. This exact

moving video background was used in the actual program, however, fonts button sizes, and text background and layout have changed.

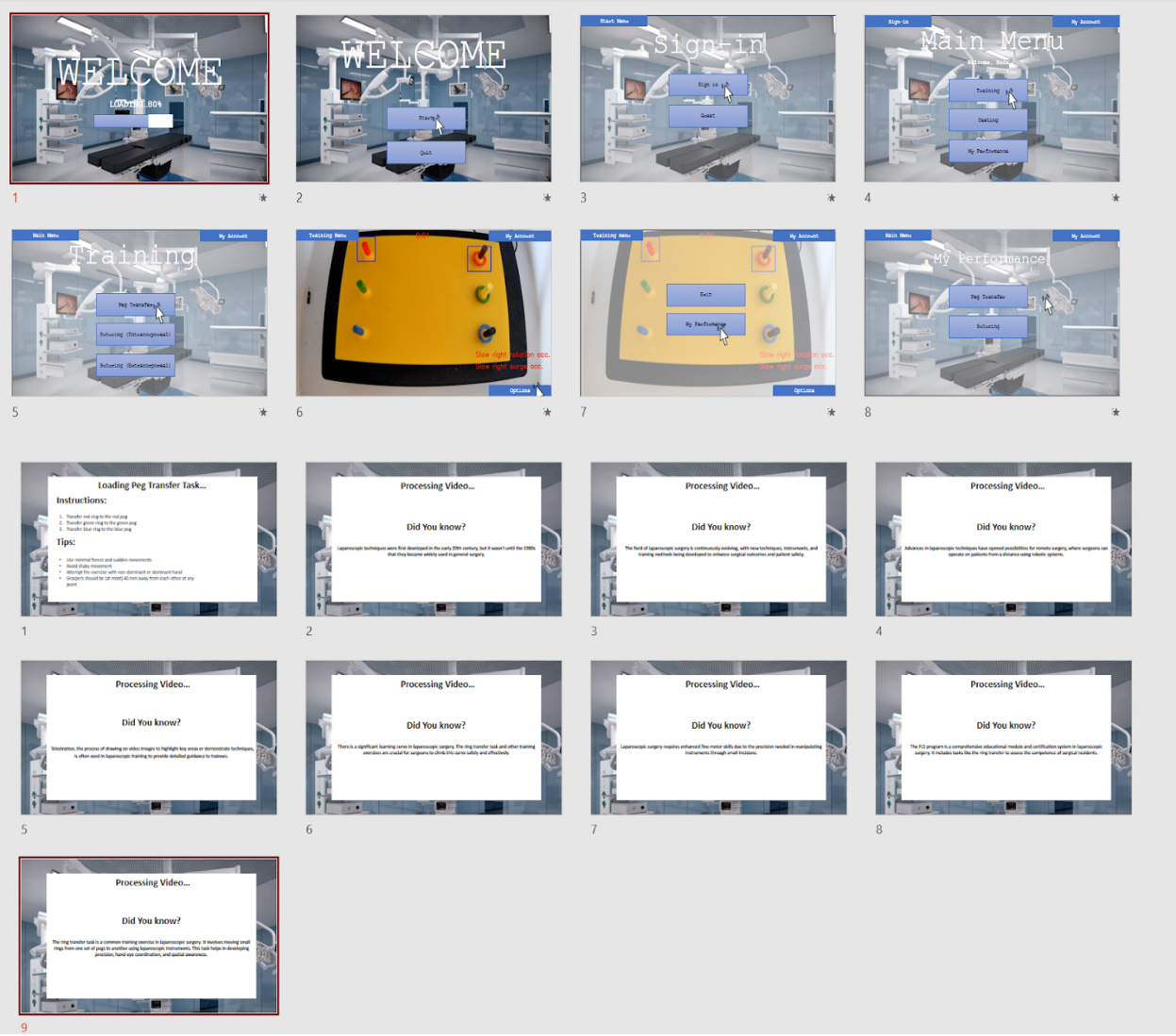


Figure 12: Sample prototype for our user interface showing various menu screens (top) and loading screens (bottom) for the first iteration of our design.

4.4.1.2. Code Hierarchy

Appendix C shows a UML class diagram of the code, which is organized into a main class and its dependencies. Class ‘Application’ runs the main window of the program which the user interacts with most of the time. It is the ‘face’ of the user interface. Class ‘LoginRegister’ takes

care of user login, sign-in, and registration capabilities. The ‘TaskExecution’ class implements the sensor data capture and camera functionalities for the three laparoscopic tasks the application currently supports. This class takes care of launching a training task, copying sensor data into a text file for subsequent data processing, and also opens or closes the camera view for a particular task depending on the program’s state. The rest of the classes deal with data processing, task segmentation, and/or performance evaluation. The ‘SensorDataCleaner’ class is responsible for sorting through the sensor data text file and converting it to a .CSV file. The ‘DataSegmentation’ class segments the data acquired into three subtasks for the analysis process to follow. The ‘TaskPerformanceAnalyzer’ class executes the algorithms responsible for performance comparison and evaluation, as detailed in Sections 4.2 and 4.3. The ‘DataPlotter’ class aligns the user’s data signals with the expert's data signals to present them on the same plot for visual performance evaluation. Finally, the ‘VideoPlayer’ class aligns videos of the user and expert performing the task so they can be viewed and compared side by side when prompted. Some classes have to open a sub window derived from the main window to implement their intended functionality. Subsequent iterations of the user interface will add other classes as more features are integrated with the simulator’s software. This structure for the user interface has provided a good foundation for future development.

4.4.1.3. Design flow of new user interface

Figure 13 shows the design flow for some of the multiple screens and windows that were developed for the program. These include the welcome screen, login/registration screen, the main menu, the task menu, the loading screen, the camera view screen, the feedback menu, the evaluation menu, the video playback menu, the history menu, the video feedback menu, the visual feedback menu, the tool trajectory menu, the subtasks menu, etc.

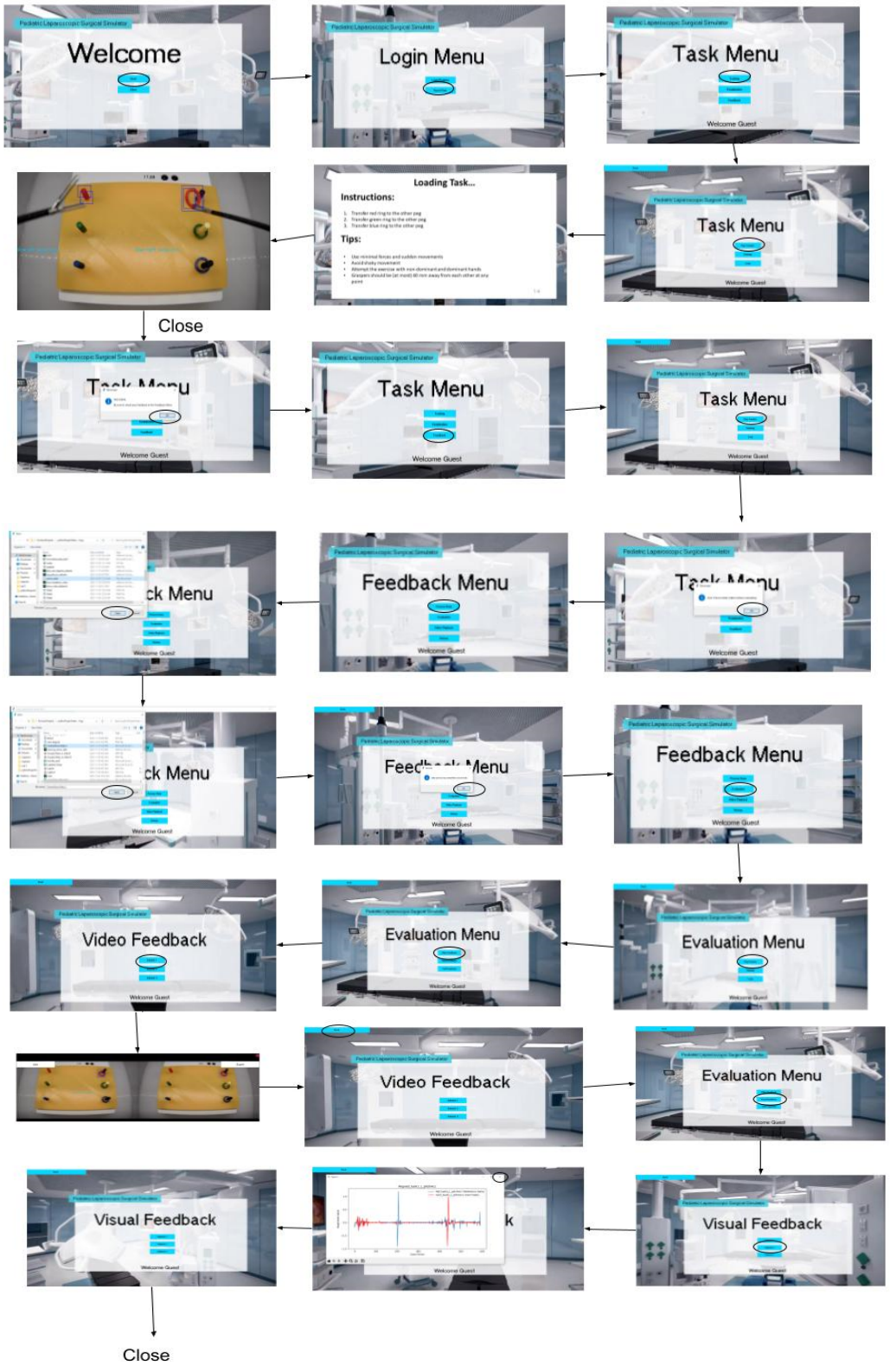


Figure 13: Design flow of new user interface shown with windows/screens that appear when certain buttons are clicked.

The welcome menu leads the user to the login menu where they can choose to login as a user, login as a guest, or register themselves. Once they are signed as a guest or authenticated user, they will be led to the main menu. The main menu allows the user to choose between the training, examination, and feedback menus. Training mode will allow the user to practice with direct feedback from the program while they perform the task. Examination mode, not shown in Figure 12 above, is identical to the 'Training' path, but tests the user's abilities and provides no indication of crossed thresholds or direct feedback to help the user. The training and examination menus will lead the user to a task menu where they can choose which task they want to perform. The task menu allows the user to choose between the peg transfer, suturing, or ligating loop tasks. Once they click on any one of these tasks, the program will launch the loading screen which gives the user an overview of the task to be performed. After 16 seconds or so for sensor calibration, the camera view will pop up and the user can immediately begin the task. Once they finish, they can close the camera view, and will be prompted to check their feedback. When they return to the main screen on the main window, they can lead themselves to the feedback menu, where they select which task, they wish to view the feedback for. From there, they'll be sent to another screen of the feedback menu where they can process their data, access evaluations of their performance, watch a video playback of the task they just performed, or access historical data saved within their directory. Clicking on the video playback button will open a sub-window where the user can play the video of the task they just performed. Before the user clicks on the 'evaluation button' on the feedback menu, they must process their data, and this instruction is given to the user before they enter the feedback menu. Once they process their data, and go to the evaluation menu, they have a few options. By clicking on the 'visual evaluation' button, they can select a given subtask of the procedure, and then view a visual evaluation of their performance in different subtasks compared

to a referenced expert. This consists of a visual overlay of the signals pertaining to the user and the expert's for different motion-related parameters. This is shown as a sub-window that allows the user to flip through the different graphs of motion-related parameters. The 'video evaluation' button allows the user to view videos of their subtasks and an expert's at the same time side by side. Although it has not been implemented yet, the 'tool trajectory' button will open a window that will show the user's tool trajectory throughout different subtasks of a procedure, compared to a referenced expert's. As seen in Figure 12 above, while buttons are used to seamlessly transition between the multiple screens, the design flow can get complicated very quickly, especially when new features are being added.

4.4.2. Development and integration of new features in the user interface

4.4.2.1. New graphics and Tkinter elements

Based on the pre-developed prototype of the graphical design discussed previously, features of Python's tkinter's library like text boxes, buttons, and threading capabilities for moving video backgrounds were used to create the program. The moving video background of surgical room equipment along with the blue tkinter buttons were used to develop a theme that mimics the environment of an actual operating theater. This theme was promoted throughout the whole program for consistency and professionalism. This allowed for code duplication and reusability, which made the development of new screens and the integration of new code much easier and more efficient to manage.

4.4.2.2. User log-in/registration feature

Figure 14 above presents the new log-in/sign-in, and registration feature that will be used for users to create an account or log into one, in order to save their historical and current data in one folder within the program directory. This presents a way for older data to be saved, so a user's

past and current performance over time can be tracked and evaluated more extensively, if they so choose. Every time someone registers, a new folder for them is created within the program directory if they do not already have an account. If the user signs in successfully, their data will be saved to the folder where their older data was saved. If a user wants to use the program as a 'guest user', their data will be saved to the program's guest folder. The login functionality will be further developed in future iterations to ensure proper storage, privacy, and protection of user data.

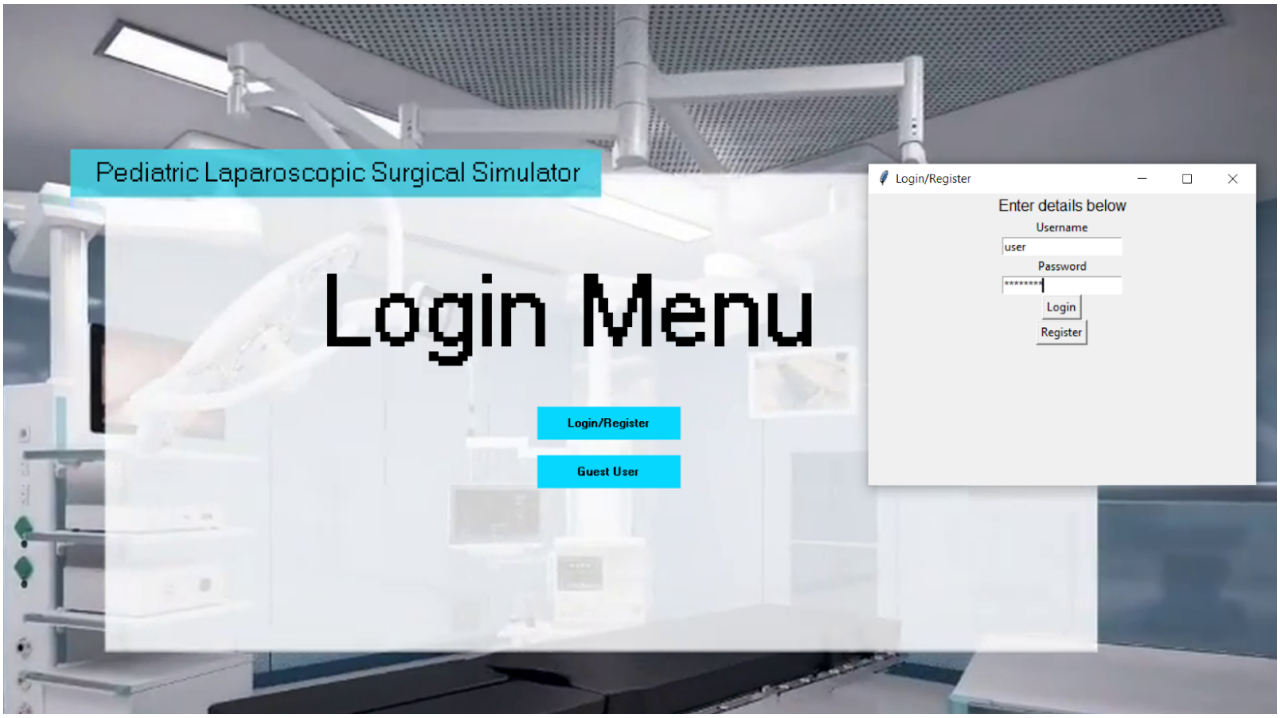


Figure 14: Login/registration screen with associated pop-window.

4.4.2.3. Integration of video playback feature log-in/registration feature

In the latest enhancement of our user interface, we have incorporated a pivotal feature: a media player script named "Video Playback." This innovative tool is expertly designed to allow users to load and review their procedure videos post-training, providing an invaluable performance analysis resource.

Developed using Python, the script is anchored in the Tkinter library, renowned for its robust graphical user interface (GUI) capabilities, as shown in Figure 5. The interface, aptly titled

"Performance Review Player," is designed for user-friendliness and efficient interaction. It includes a 'Load' button, enabling users to upload their video files effortlessly. This is complemented by the 'Play/Pause' button and additional controls that allow users to skip through the video, enhancing the maritime experience.

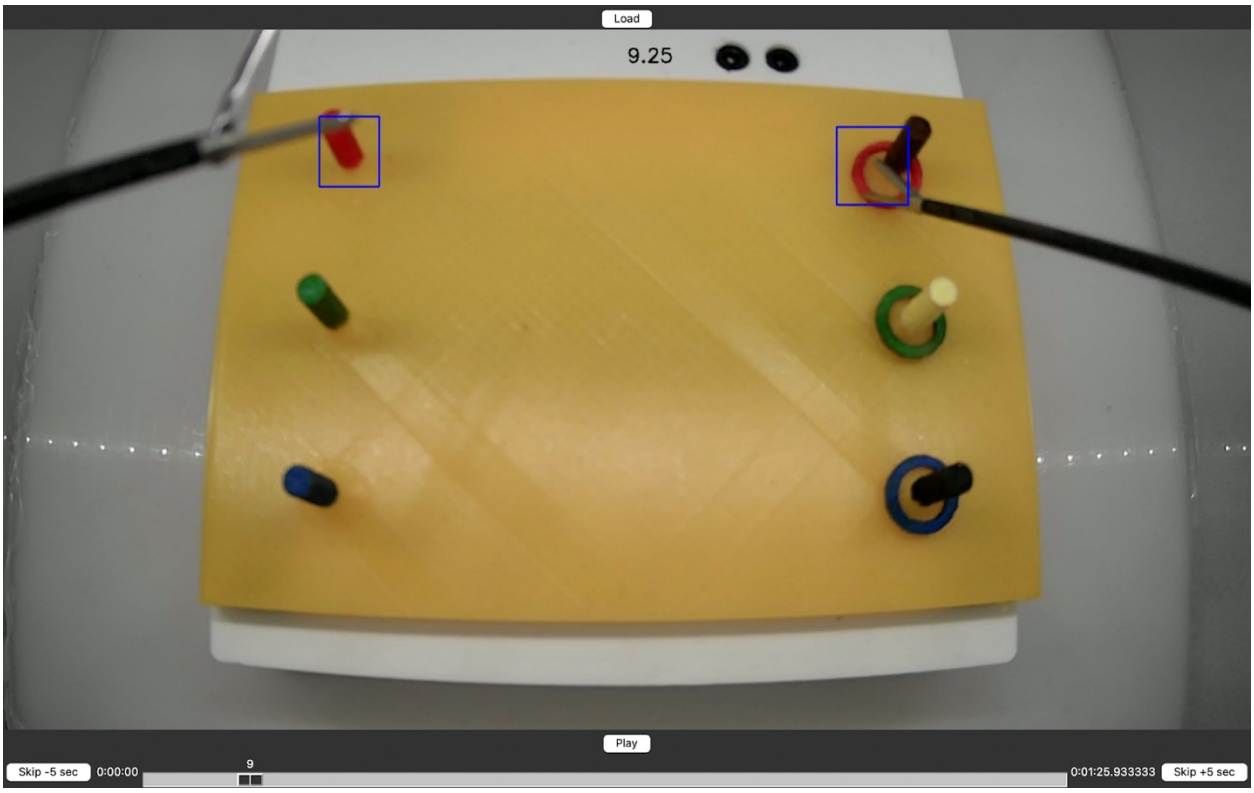


Figure 15: A media player application for procedure videos post-training.

The script's heart lies in its video playback functionality, handled by the TkinterVideo component. This component not only ensures seamless video display but also supports scaled playback within the interface, adapting to different user needs. The progress slider is a standout feature, offering users the ability to track their progress through the video and to seek specific segments effortlessly. This functionality is further enriched by dynamically updating start and end time labels, clearly and concisely representing the video's timeline.

Event handling is meticulously managed within the script, with custom functions addressing user interactions such as video loading, seeking, and skipping. These functions are

pivotal in rendering the playback experience both responsive and intuitive. Additionally, the script is adept at detecting when the video has ended, resetting the playback controls and slider to their initial states, thereby ensuring a smooth user experience.

Executed within the main block, the script initializes the VideoPlayerApp class and activates the Tkinter main loop. This initiation is crucial as it keeps the graphical interface responsive and engaging for the user.

Overall, the "Video Playback" feature is a significant stride in enhancing our interface's usability. It simplifies the review process for users and is crucial in facilitating their learning and improvement, marking a significant milestone in our commitment to delivering user-centric solutions.

4.4.2.4. Integration of new data processing procedure

In the previous implementation of the user interface, the sensor data was processed behind the scenes whenever the user wanted to view their performance for different motion-related parameters throughout a procedure. The processing only consisted of reading the sensor data from the text file, and then simply graphing it for the user.

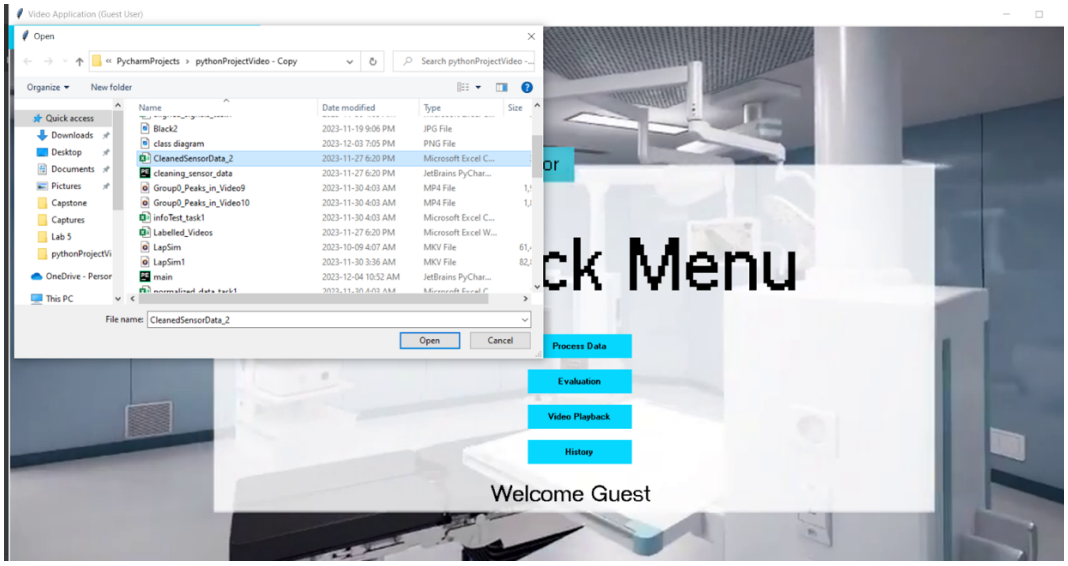


Figure 16: Opening of file explorer menu upon clicking ‘Process Data’

However, the new script allows the user to select their collected sensor data text file from the directory once they click the ‘Process Data’ button, as shown in Figure 16. This data is cleaned and transformed into a .CSV file. It is then labeled and segmented into gestures as described in Section 4.3, and the user is prompted with a pop-window that informs it to pick the correct file with the labeled data in order to prepare for the algorithmic steps that will follow later.

This new data processing procedure was integrated into the new user interface through the inclusion of the relevant classes, as shown in the class diagram in Appendix C. The relevant buttons were linked to the corresponding functions in the Application class, which instantiated the classes required for data processing.

4.4.2.5. Integration of new performance evaluation features

The previous implementation did not include any way for a user to compare their performance to an expert in a direct and specific manner. No reference data for an expert was given, besides the indications of simple motion thresholds that were crossed in the procedure.

Continuing on from the previous section, after the user selects the labeled procedure file, they must wait for around 2-3 minutes while the program segments the procedure into three subtasks and generates parameters for algorithms. These algorithms will allow for comparison of an expert’s signals to the user’s signals for various motion-related variables, both graphically, and visually through videos. A loading screen will pop up to entertain the user while they wait. Once the program finishes, the user can click on the ‘Video Feedback' or ‘Visual Feedback’ buttons to view their graphical or video feedback for the three subtasks of the procedure. A sample of the feedback the user could receive is shown in Figure 17 below.

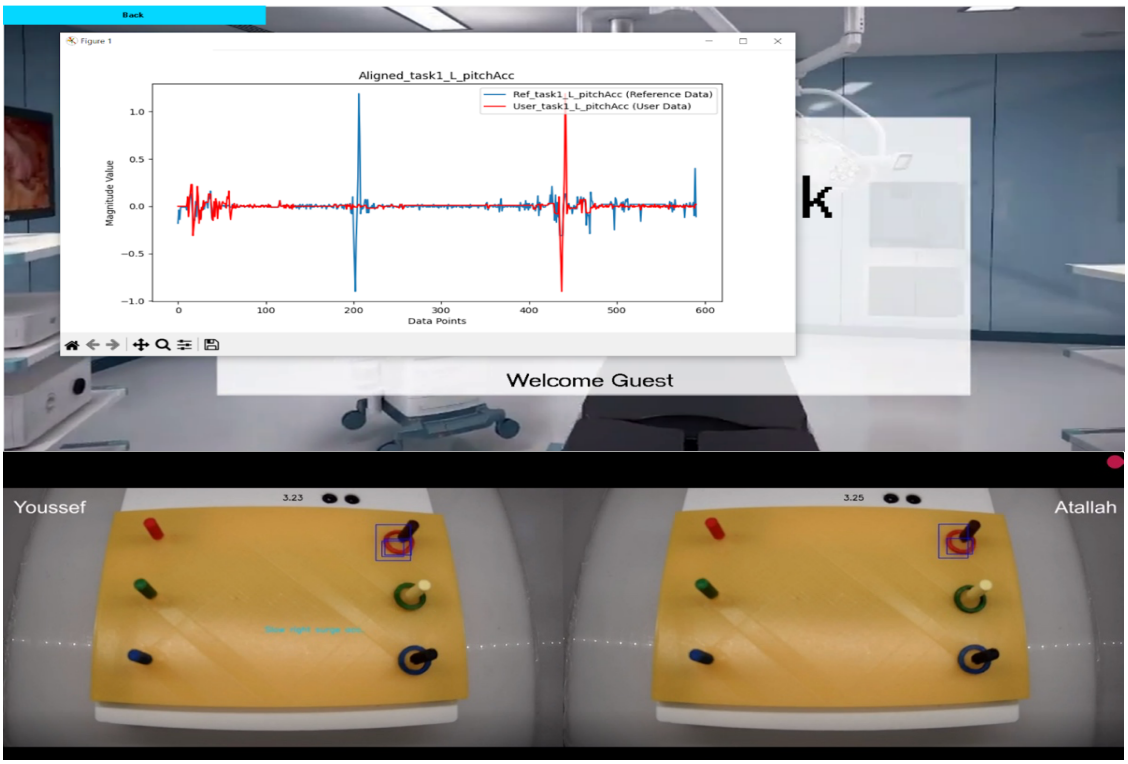


Figure 17: Integration of enhanced visual (top) and video (bottom) feedback

This new procedure for performance evaluation was integrated into the new user interface through the inclusion of the relevant classes, as shown in the class diagram in Appendix C. The relevant buttons were linked to the corresponding functions in the Application class, which instantiated the classes required for running the algorithms related to performance evaluation.

4.5. Training Tasks for Laparoscopy

The previous implementation of the pediatric laparoscopic simulator only set up the peg transfer task. We wanted to add more sophisticated training tasks like the suturing and ligating loop tasks. Currently, we have not yet implemented any other tasks besides the peg transfer task as we were preoccupied with setting up a good foundation for the hardware and software of the simulator from which we can work off of. For now, only the peg transfer task works, exactly as it was before. The other tasks have not been set up yet, however, the new designed force-sensor task platform has been properly designed to allow for the introduction of another training task.

4.6. Object Detection

Timely and accurate feedback is crucial for improving the efficacy of surgical training. However, the traditional approach, which relies on manual evaluation for feedback, can be subjective, inconsistent, and resource intensive. Additionally, manually labelling surgical gestures in training videos for feedback purposes is time-consuming and prone to human error. This limitation restricts delivering immediate, objective, and detailed performance analysis. The reliance on manual approaches to evaluate surgical training activities poses a substantial barrier to reaching the highest standards of surgical proficiency. This approach must be adjusted to maintain training efficiency and feedback quality. There is a clear need for a technology-driven solution that enhances the accuracy and reliability of performance assessments in surgical training by streamlining the feedback mechanism.

The proposed solution is to automate the gesture labelling process using state-of-the-art object detection techniques. This automation aims to replace manual gesture labelling, traditionally performed by experts, and then recorded in spreadsheets for analysis. The method will specify, classify, and label different surgical gestures from training videos using deep learning-based object detection methods. This process improves precision and consistency in gesture recognition and significantly reduces the time and effort involved in manual labelling.

To this end, our task is to develop an automated system using deep learning that can accurately recognize and label surgical gestures in real time. This system will analyze surgical training videos, identify specific gestures, and label them without human intervention. Secondly, automated labelling should be used to evaluate trainees and provide them instant feedback quickly and objectively. The system will automatically analyze the trainees' performance, recognize their

gestures, and automatically generate feedback information. This feedback will be systematically recorded in a structured format, such as an Excel sheet, for easy access and analysis.

The two main objectives guided this task are:

- Developing object modelling specifically tailored for training tasks.
- Implementing a deep learning-based object detection method in our surgical simulator can support real-time applications.

Through this approach, the aim is not only to enhance the accuracy and efficiency of our model but also to contribute innovative solutions in object detection and tracking to the more comprehensive field of surgical training.

4.6.1. Object Modelling

In creating the object model for the machine learning application, the focus was on describing the critical elements required for the surgical training simulations, such as identifying the left and right graspers and the rings and their various states and movements. To accomplish this, we outlined two fundamental attributes for each of these objects:

- Location: This attribute precisely describes the grasper's pixel coordinates in the simulation to ensure proper spatial tracking.
- Movement: We recognized movement as an important relational attribute that captures the interactions between objects, such as holding the ring together with the left grasper, which indicates a movement of "Ring transfer. This attribute is required to recognize each action's purpose and context within the simulation.

As described in Table [2], the object model demonstrates a comprehensive list of potential movements and states. These include the idle state of the graspers (LG0 and RG0), various activities towards and away from the operating table (e.g., LG1, RG1), and interactions with the

rings, such as clasping or transferring (e.g., LG3, RG6). This table provides a structured framework for the object's potential movements and relations within the simulation. Further, we used a semantic object model approach, which augments the essential attributes of location and movement with their semantic implications. This increases the machine learning algorithm's ability to study the activities within the simulation, moving beyond mere coordinate tracking to understanding the semantic purpose of each movement. For instance, the semantic model allows the algorithm to determine that "LG3": Left pick ring: Clasping ring represents an action where the left grasper picks and clasps the ring instead of identifying the spatial coordinates of the grasper and ring. [26] The semantic object model facilitates object state detection by focusing on location, making it more efficient. This approach allows our machine learning model to accurately track location and analyze contextual relationships in surgical training simulations for real-time feedback and assessment.

Table 2: The Objects Classification by Movement and Location.

Gesture Index (Right) and (L: Left)	Objects Movement	Jaws (Open or Closed)
RG0	Ring on peg: Idle state (or hovering)	C
RG1	Right grasper: Moving vertically downwards towards peg on right side of operating table	O/C
RG2	Ring out peg: Positioning ring between the jaws and clasping it	O
RG3	Right pick ring: Moving upwards with clasped ring	O
RG4	Right Carry ring: Moving to center of operating table to meet left grasper	O
RG5	Right carry ring: Hovering with ring between jaws for left grasper to pick	O
RG6	Ring transfer: Holding ring together with left grasper	O
RG7	Right pick ring: Opening jaws to let go of ring	O
RG8	Moving aside	O/C
LG0	Idle state (or hovering)	C
LG1	Left grasper: Moving to center of operating table to meet right grasper	O/C
LG2	Left pick ring: Positioning ring between the jaws	O
LG3	Left pick ring: Clasping ring	O
LG4	Left carry ring: Moving horizontally (straight or diagonally) towards a peg on left side of operating table	O
LG5	Left carry ring: Positioning ring over peg	O
LG6	Ring on peg: Opening jaws and dropping ring through peg	O/C
LG7	Ring on peg: Moving aside after transferring all three rings	O/C

4.6.2. Deep Learning: Single Shot Detection (SSD)

After setting up a comprehensive object model, our next task is implementing object detection using deep learning, primarily via Single Shot Detection (SSD). A single-shot detector

is a one-phase detector that predicts multiple classes. In other words, a real-time object detection framework that merges the tasks of object localization and type into a single unified architecture [27].

This section will discuss the training process, including data preparation, annotation, and fine-tuning hyperparameters to optimize the model's performance for our specific use matter. Challenges encountered, methods used to overcome them, and the initial results will also be presented, delivering a clear view of the progress made and the steps forward.

4.6.2.1. Dataset Collection and Preparation

For this task, we have created a specialized dataset from laparoscopic training videos, each ranging from 75 seconds to 2 minutes. This dataset includes frames extracted from these videos, carefully annotated to function as a training foundation for our SSD model. The annotation process was facilitated using LabelImg, an open-source image annotation tool [28], which enabled the labelling process and resulted in each image having an associated .xml file. These .xml files were transformed into .csv files for an overview and. tfrecord files for use in the training pipeline. Upon completing the annotation process, we classified the images and their corresponding .xml, .csv, and. tfrecord files into separate datasets for training, testing, and evaluation, following a 70:20:10 split. This split is created to deliver a complete training experience while ensuring a thorough validation of the model's effectiveness.

4.6.2.1.1. Label Map

A required component of the dataset preparation was preparing a label map. This map is a directory for the training and detection algorithms, linking each object class to a specific integer value. For the dataset, we formulated a label map with 15 unique labels; each label corresponds to a particular laparoscopic peg transfer gesture, such as 'RG1' for the first gesture involving the right

grasper, 'LG1' for the left grasper, and so forth, as shown in Figure [18] These labels are instrumental for the SSD model to recognize and categorize the precise gestures performed during the laparoscopic procedures. Then, the label map is saved in a. pbtxt file format and is required to train the SSD model to accurately identify and classify the different instruments observed during detection. Figure [18] illustrates an example of our label map configuration.

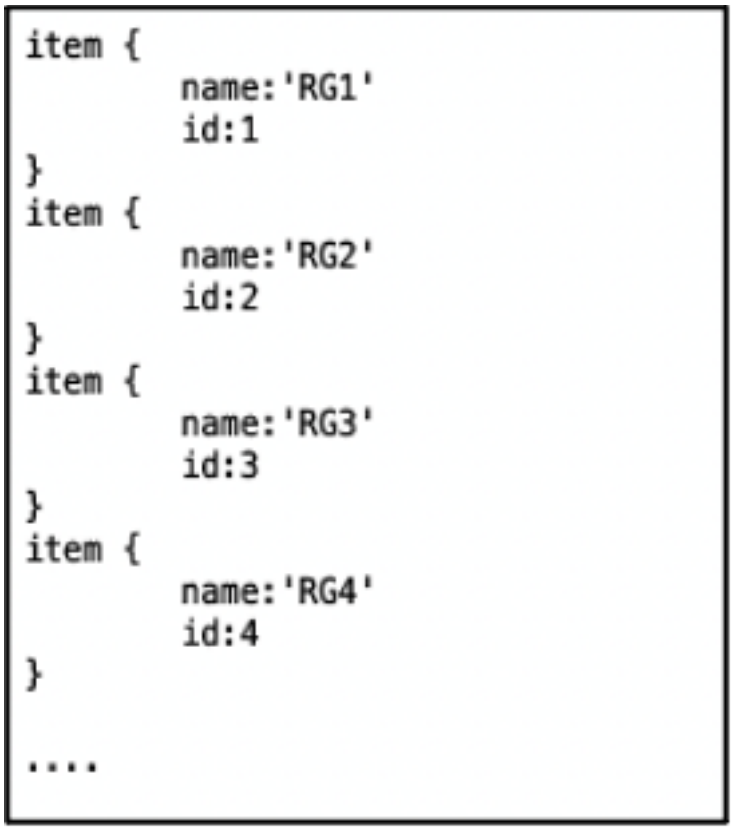


Figure 18: Label map file classes.

4.6.2.1.2. Data Augmentation

We augmented our dataset utilizing the TensorFlow Object Detection API Image Preprocessor tool to train the object detection model. This raised variations to our dataset without compromising the quality of the original data. Our data augmentation pipeline included modifications such as unexpected horizontal flipping and adjustments to brightness and contrast. These measures trained the model to identify and classify objects under various conditions. Using

these TensorFlow API data augmentation variables, we expanded our dataset effectively, improving the model's robustness and capability to generalize to unknown images [29].

4.6.2.2. Model Architecture

For our model's foundation, we selected the SSD MobileNet v2 320x320, pre-trained on the COCO 2017 dataset, from the TensorFlow 2 Detection Model Zoo [29]. This model is a useful starting point for training on our dataset, effectively good speed, and precision for real-time detection tasks. The architecture, shown in Figure [19], includes a feature pyramid network (FPN), improving the model's ability to accurately detect objects across different scales.

The SSD MobileNet V2 architecture uses depthwise separable convolutions—a computationally efficient option to standard convolutions—allowing for faster processing without a substantial loss in accuracy. This feature is important for real-time applications, as our project required immediate surgical gesture detection. As the model processes input frames, such as the 'Input for RG1' displayed, it generates bounding boxes at multiple scales, pinpointing the location of surgical gestures within the frame.

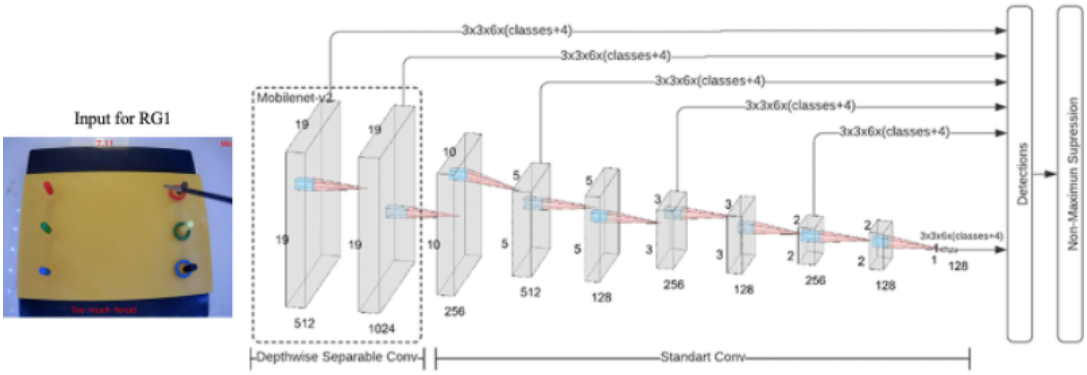


Figure 19: Real-time Surgical Gesture Recognition with SSD MobileNet v2 320x320 Architecture [29].

4.6.2.3. Model Evaluation

After training the SSD MobileNet v2 320x320 model, we used TensorBoard, a visualization tool within the TensorFlow Object Detection API, to observe and evaluate different

metrics, ensuring the model's performance was fine-tuned to our specific requirements. TensorBoard provided us with detailed insights into the model's detection abilities through different metrics:

A. Precision and mAP (mean Average Precision): The precision of our model was observed at different Intersections over Union (IoU) thresholds. The TensorBoard plots, as shown in Figure [20], that our model showed a progressive increase in precision across different object sizes—small, medium, and large, especially in medium-sized object detection as shown in DetectionBoxes Precision /mAP(medium) Figure [20], which is essential for accurately determining surgical gestures of various dimensions.

B. Recall Metrics: allow us to understand how well the model can determine all the dataset's suitable cases. As shown in Figure [21], the Average Recall (AR) remained constant across the board, showing a stable detection rate throughout training iterations. This consistency is important to providing reliable gesture detection during surgical training.

C. Loss Metrics: Figure [22] shows that the classification and localization loss metrics indicate a decline, demonstrating a model's improving accuracy in classifying and localizing objects. The regularization loss remained relatively flat, suggesting that the model was not overfitting to the training data.

These evaluations are important for understanding the model's stability and limitations, and the visualizations from TensorBoard allow us to iterate on the model to optimize its performance further.

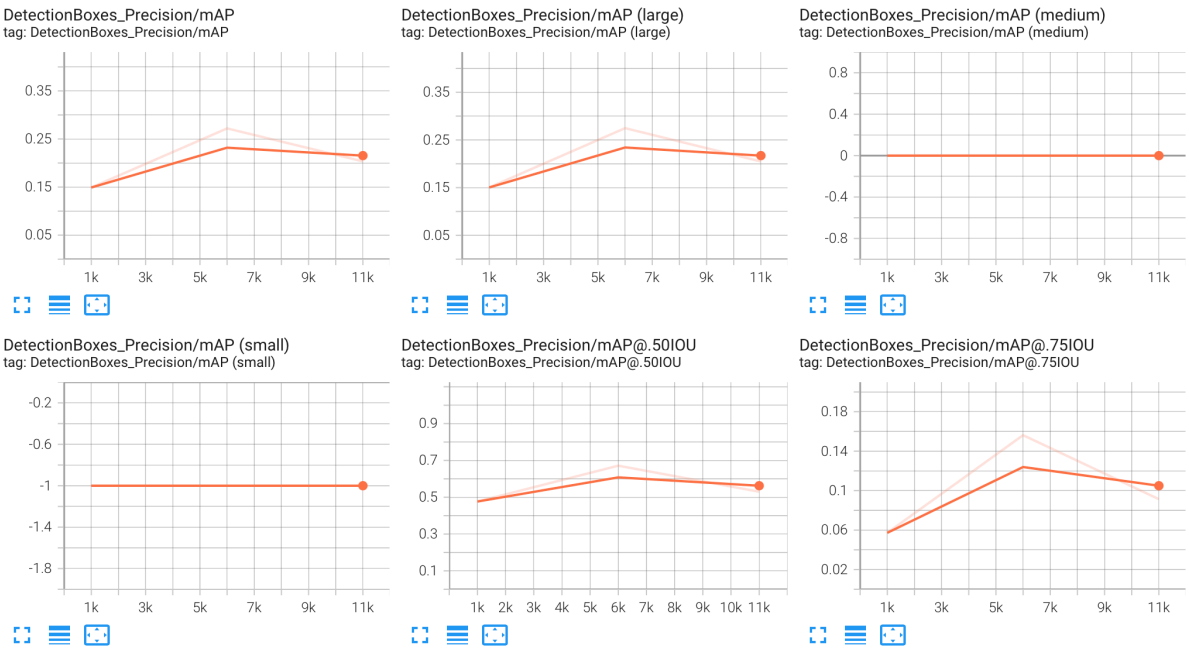


Figure 20: Model Precision Analysis: Mean Average Precision (mAP) Across Different Object Sizes and IOU Thresholds.

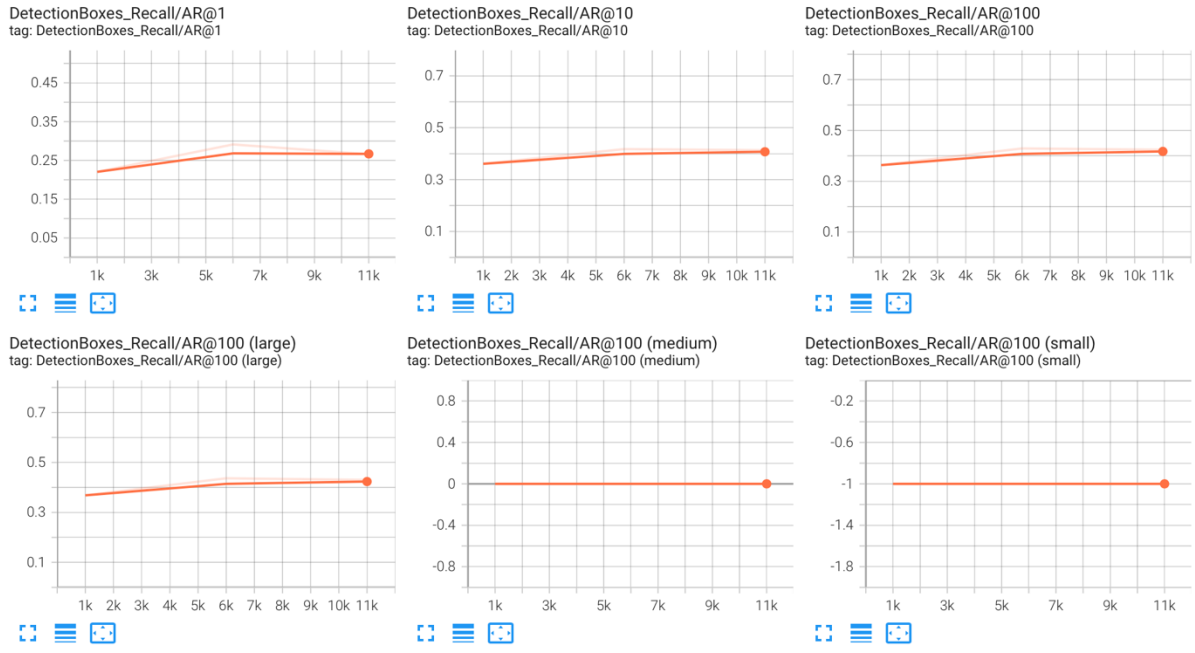


Figure 21: Model Recall Assessment: Average Recall (AR) at Multiple Detection Thresholds.

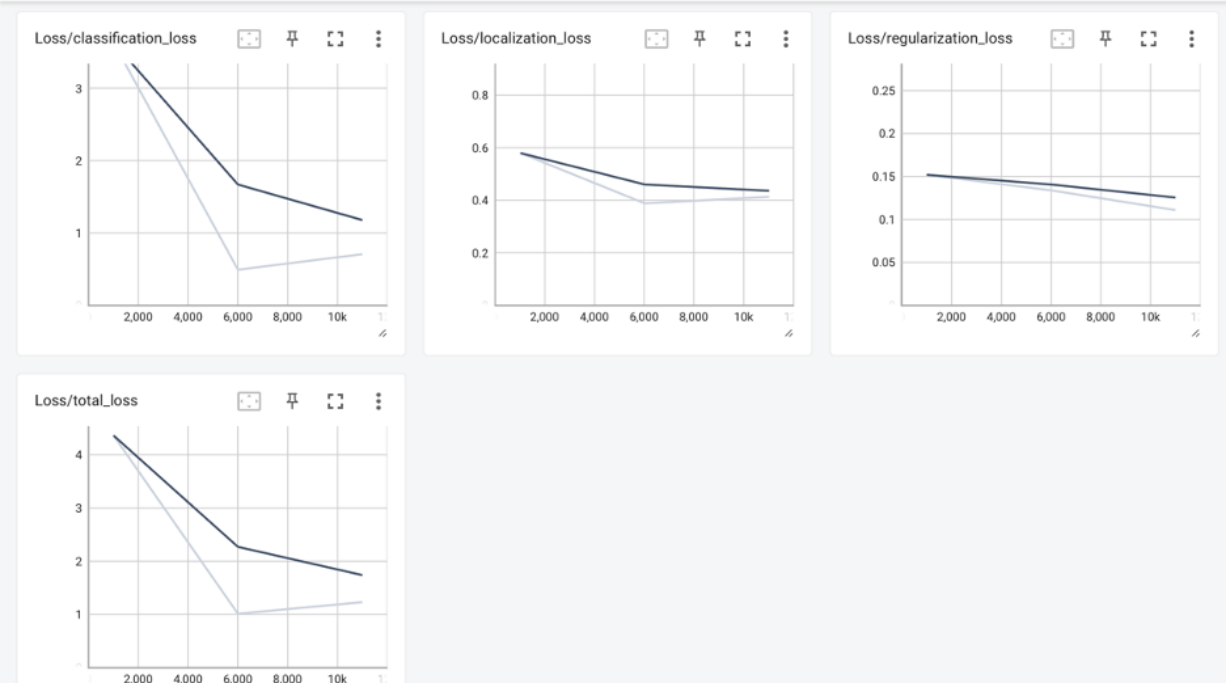


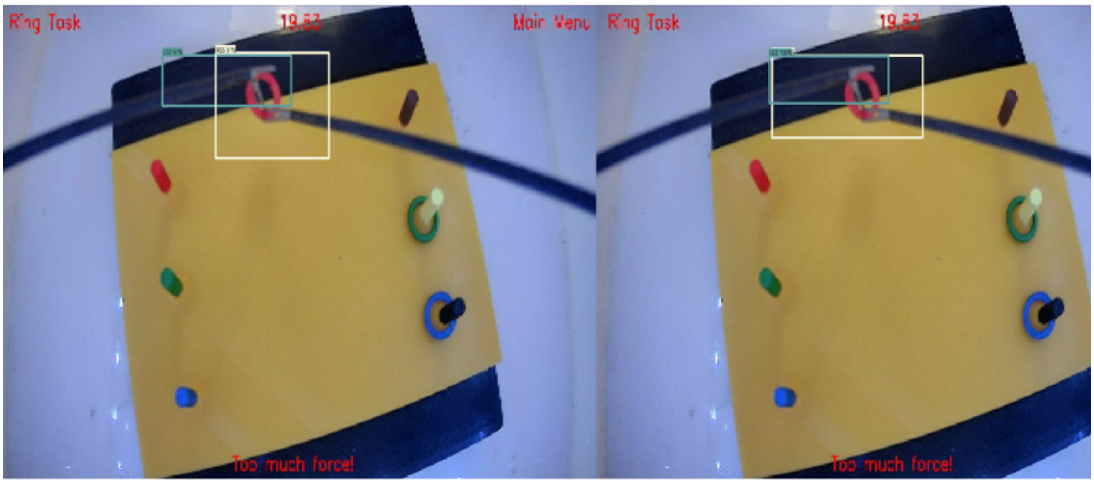
Figure 22:The Analysis of Loss Metrics Over Iterations.

4.6.3. Evaluation

During the testing stage of our SSD MobileNet v2 320x320 model, we developed a detailed script to automate the detection and labelling of surgical gestures from video data. The script loaded the pre-trained model, processed each video frame, and applied the detection function to identify gestures within the footage. Whenever a gesture was detected with adequate confidence, the script recorded the event in a CSV file and its timestamp, creating an organized log for performance analysis.

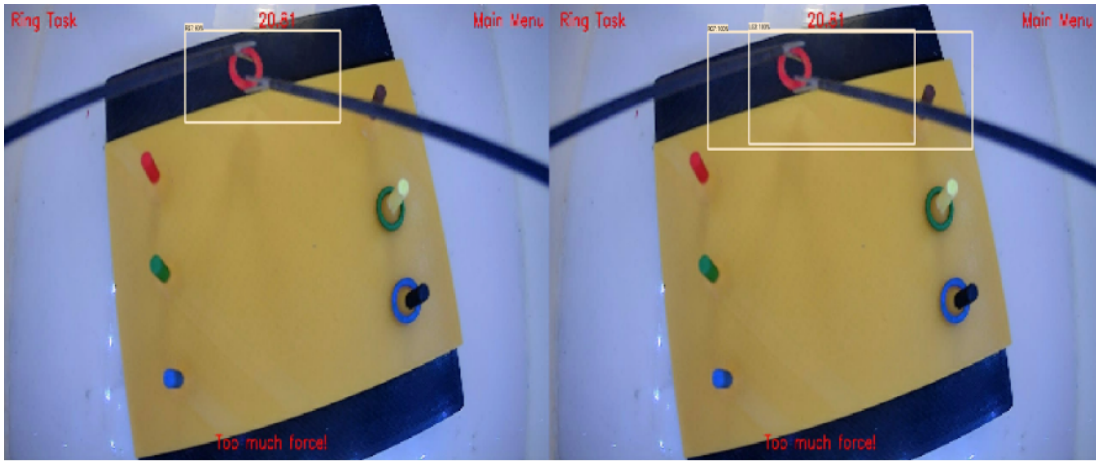
After implementation, the model was evaluated for its ability to recognize and label various gestures, including surgical gestures. As shown in both Figures [23] and [24] Image A represents the system's output post-training, showcasing detected gestures in real-time. Image B, however, displays the gestures that are ideally labelled after manual review and labelling, which acts as a reference for what the system is expected to achieve, as shown in the annotated test images Figure [23], gestures were identified with high confidence, such as 'LG2' and' RG5' being labelled with

91% confidence. However, lower confidence detection and non-detection were observed, as shown in Figure [24], which indicates that 'RG7' with 60% confidence and 'LG3' is not presented, suggesting a necessity for further model improvement. These discrepancies are likely due to the limited diversity within the training dataset, implying that increasing the dataset could improve the model's accuracy. Despite these challenges, the automated system's potential for providing real-time feedback in surgical training environments remains clear.



(a) (b)

Figure 23: Comparative Analysis of Automated Accurate Gesture Detection: SSD MobileNet v2 320x320 vs. Manual Labeling.



(a) (b)

Figure 24: Comparative Analysis of Automated Inaccurate Gesture Detection: SSD MobileNet v2 320x320 vs. Manual Labeling.

5. Future Planned Improvements

The user interface will undergo multiple iterations until all new software and hardware features have all been implemented. First of all, with the integration of new tasks like the suturing and threading tasks, new code will be needed to assess the user's execution of the task through a camera view, and additional screens and functions will also need to be added. Other elements of the signal and data processing steps still need to be integrated into the user interface as well, which will require additional organization or re-factoring of the code. File paths and user-related variables will need to be defined globally to ensure the program knows which user is signed throughout the duration of the program and saves the user's data in the correctly specified location.

In addition, improved feedback can be given while the user is performing a task, like the use of beep alarm sound to indicate when they have crossed certain parameters, like exceeding the limits of the task's workspace. We are also planning on enhancing the simulator by integrating a feature that displays a realistic range of the procedure. This will be achieved through a visual bar representing the distance between the surgical tools, specifically the graspers. This bar provides immediate feedback on the spacing used during the procedure. The colour coding of the bar is intuitive and informative: red signifies that the tools are nearing the edge of the camera's view with the clear beep sound that was mentioned, indicating a potential loss of visibility. Yellow represents a moderate range, alerting the user to adjust accordingly, while green indicates the optimal spacing range. This ideal range, measuring 40x40 mm, was determined in consultation with two experienced surgeons we collaborated with earlier in the semester, Dr. Ahmed Nasr and Dr. Georges Azzie. Their input was instrumental in ensuring the realism and educational efficacy of the simulator. Regarding performance evaluation, while the user is given feedback in visual and graphical form, it is important to translate this feedback to the user, outside of the mathematics

and mechanics of the procedure. For instance, this can mean defining what happens during each subtask, and what the user should do to improve their performance. It will be important to present feedback received from the performance evaluation algorithms in a more understandable form.

Additionally, in order to improve the precision and efficiency of our object detection model, we are analyzing different pre-trained models, including the SSD-ResNet50 V1 FPN architecture. This architecture is expected to offer a more resilient feature extraction, which will assist in detecting nuanced gestures with greater accuracy[31]. Furthermore, we intend to enlarge our dataset by incorporating a more comprehensive array of gesture variations and labelling them. This will allow the model to learn from a more diverse and inclusive sample while enhancing its generalization abilities. Finally, a final re-factoring and cleaning of the code will be needed to ensure efficient and smooth execution.

6. Report Contributions

Table 3: Outline of each member's contribution to the proposal report

Contributor	Section
Atallah Madi	• Introduction • Progress Description (Hardware Improvements) • Appendices
Esraa Alaa Aldeen	• Progress Description (Object Modelling) • Future Planned Improvements
Huda Sheikh	• Progress Description (User Interface (UI) Design and Training Tasks for Laparoscopic) • Future Planned Improvements
Youssef Megahed	• Introduction • Progress Description (Data Processing Pipeline, Evaluating Trainee's Performance Using Dynamic Time Warping (DTW), and Development, and integration of new features in the user interface) • Future Planned Improvements

References

[1] "Laparoscopic Surgery - What is it? | ASCRS," *fascrs.org*. https://fascrs.org/patients/diseases-and-conditions/a-z/laparoscopic-surgery-what-is-it

[2] "What Tools are Used in Laparoscopic Surgery | Laparoscopic.MD," https://www.laparoscopic.md/surgery/instruments#:~:text=Laparoscopic%20instruments%20are%20made%20of.

[3] "Laparoscopy in Children," *myhealth.alberta.ca*. https://myhealth.alberta.ca/health/Pages/conditions.aspx?hwid=acg9455

[4] Bojan Gavrilović, A. S. Fahy, B. Carrillo, A. Nasr, J. Ted Gerstle, and Georges Azzie, "Development of an Open-Source Laparoscopic Simulator Capable of Motion and Force Assessment: High Tech at Low Cost," *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 28, no. 10, pp. 1253–1260, Oct. 2018, doi: https://doi.org/10.1089/lap.2018.0126.

[5] P. K. H. Tam, "Current topic: Laparoscopic surgery in children," *Archives of Disease in Childhood*, vol. 82, no. 3, pp. 240–243, Mar. 2000, doi: https://doi.org/10.1136/adc.82.3.240.

[6] E. Westwood, B. Malla, J. Ward, R. Lal, and K. Aryal, "The Impact of a Laparoscopic Surgery Training Course in a Developing Country," *World Journal of Surgery*, vol. 44, no. 10, pp. 3284–3289, Jun. 2020, doi: https://doi.org/10.1007/s00268-020-05606-y.

[7] M. Nagendran, C. D. Toon, B. R. Davidson, and K. S. Gurusamy, "Laparoscopic surgical box model training for surgical trainees with no prior laparoscopic experience," *Cochrane Database of Systematic Reviews*, Jan. 2014, doi: https://doi.org/10.1002/14651858.cd010479.pub2.

[8] Y. A. Oquendo, E. W. Riddle, D. Hiller, T. A. Blinman, and K. J. Kuchenbecker, "Automatically rating trainee skill at a pediatric laparoscopic suturing task," *Surgical Endoscopy*, vol. 32, no. 4, pp. 1840–1857, Oct. 2017, doi: https://doi.org/10.1007/s00464-017-5873-6.

[9] P. Korzeniowski, C. S. Chacon, V. R. Russell, S. A. Clarke, and F. Bello, "Virtual Reality Simulator for Pediatric Laparoscopic Inguinal Hernia Repair," *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 31, no. 11, pp. 1322–1330, Nov. 2021, doi: https://doi.org/10.1089/lap.2020.0423.

[10] G. Azzie *et al.*, "Development and validation of a pediatric laparoscopic surgery simulator," *Journal of Pediatric Surgery*, vol. 46, no. 5, pp. 897–903, May 2011, doi: https://doi.org/10.1016/j.jpedsurg.2011.02.026.

[11] H.-C. Hur, D. Arden, L. E. Dodge, B. Zheng, and H. A. Ricciotti, "Fundamentals of Laparoscopic Surgery: A Surgical Skills Assessment Tool in Gynecology," vol. 15, no. 1, pp. 21–26, Jan. 2011, doi: https://doi.org/10.4293/108680810x12924466009122.

[12] "FLS Manual Skills Written Instructions and Performance Guidelines Important Scoring Information." Available: https://www.flsprogram.org/wp-content/uploads/2014/03/Revised-Manual-Skills-Guidelines-February-2014.pdf

[13] S. Miles and N. Donnellan, "Learning Fundamentals of Laparoscopic Surgery Manual Skills: An Institutional Experience With Remote Coaching and Assessment," *Military Medicine*, Apr. 2021, doi: https://doi.org/10.1093/milmed/usab170.

[14] M. Varras, N. Nikiteas, V. Varra, F. Varra, E. Georgiou, and C. Loukas, "Role of laparoscopic simulators in the development and assessment of laparoscopic surgical skills

in laparoscopic surgery and gynecology (Review)," *World Academy of Sciences Journal*, Mar. 2020, doi: https://doi.org/10.3892/wasj.2020.41.

[15] J. J. Park, J. Tiefenbach, and A. K. Demetriades, "The role of artificial intelligence in surgical simulation," vol. 4, Dec. 2022, doi: https://doi.org/10.3389/fmedt.2022.1076755.

[16] S. M. B. I. Botden, S. N. Buzink, M. P. Schijven, and J. J. Jakimowicz, "Augmented versus Virtual Reality Laparoscopic Simulation: What Is the Difference?," *World Journal of Surgery*, vol. 31, no. 4, pp. 764–772, Mar. 2007, doi: https://doi.org/10.1007/s00268-006-0724-y.

[17] C. Sewell *et al.*, "Providing metrics and performance feedback in a surgical simulator," *Computer Aided Surgery*, vol. 13, no. 2, pp. 63–81, Jan. 2008, doi: https://doi.org/10.3109/10929080801957712.

[18] "Arduino Uno Vs Nano Vs Mega, Pinout, and technical Specifications," Electronic Clinic, Jul. 25, 2020. https://www.electroniclinic.com/arduino-uno-vs-nano-vs-mega-pinout-and-technical-specifications/ .

[19] "Strain gauge 2b.jpg - Wikipedia," commons.wikimedia.org, May 31, 2023. https://en.m.wikipedia.org/wiki/File:Strain_gauge_2b.jpg.

[20] "500g Scale calibration weight," https://modernistpantry.com/products/500g-scale-calibration-weight.html

[21] "Single point load cell H10A," BOSCHE. https://www.bosche.eu/en/scale-components/load-cells/single-point-load-cell/single-point-load-cell-h10a

[22] B. Thuraisingham, "A primer for understanding and applying data mining." It Professional, 2(1), 28-31, (2000). Available: https://ieeexplore.ieee.org/abstract/document/819936.

[23] M. Müller, “Dynamic time warping.” In: Information retrieval for music and motion, 69-84, (2007). Available: https://doi.org/10.1007/978-3-540-74048-3_4.

[24] H. Li, “On-line and dynamic time warping for time series data mining.” Int. J. Mach. Learn. & Cyber. 6, 145–153, (2015). Available: https://doi.org/10.1007/s13042-014-0254-0.

[25] N. Chauhan, “Dynamic Time Warping (DTW) Algorithm in Time Series,” the AI dream, (2022). Available: https://www.theaidream.com/post/dynamic-time-warping-dtw-algorithm-in-time-series.

[26] Brachman, R. J., & Levesque, H. J. (2009). Knowledge representation and reasoning. Elsevier.

[27] Liu, W. et al. (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science (), vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2 [28] Tzutalin. (2023). LabelImg. GitHub repository. https://github.com/tzutalin/labelImg

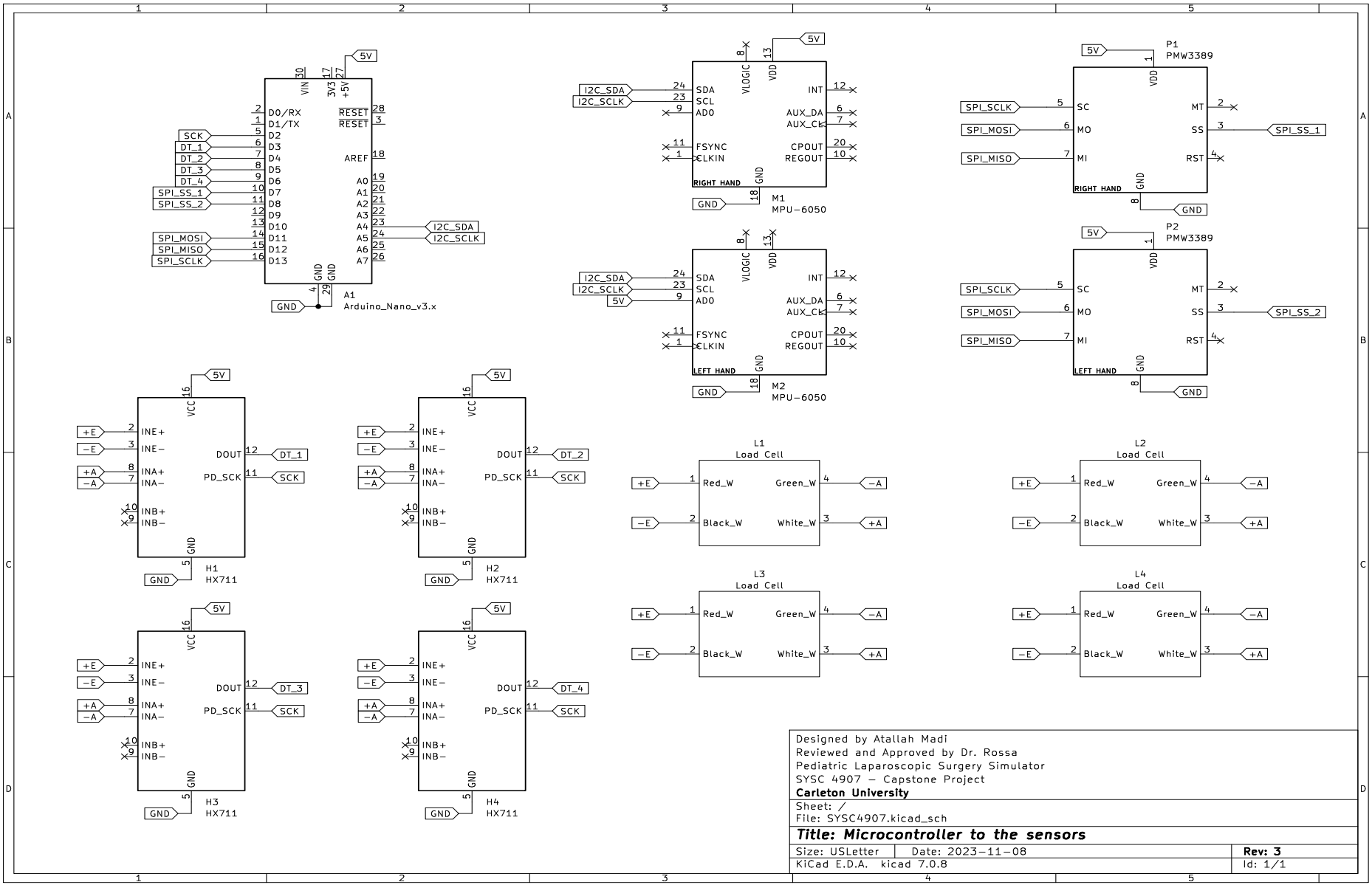
[28] Tzutalin. (2023). LabelImg. GitHub repository. https://github.com/tzutalin/labelImg

[29]] Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). "Best practices for convolutional neural networks applied to visual document analysis". In Proceedings of the Seventh International Conference on Document Analysis and Recognition, 958-963. DOI:10.1109/ICDAR.2003.1227801.

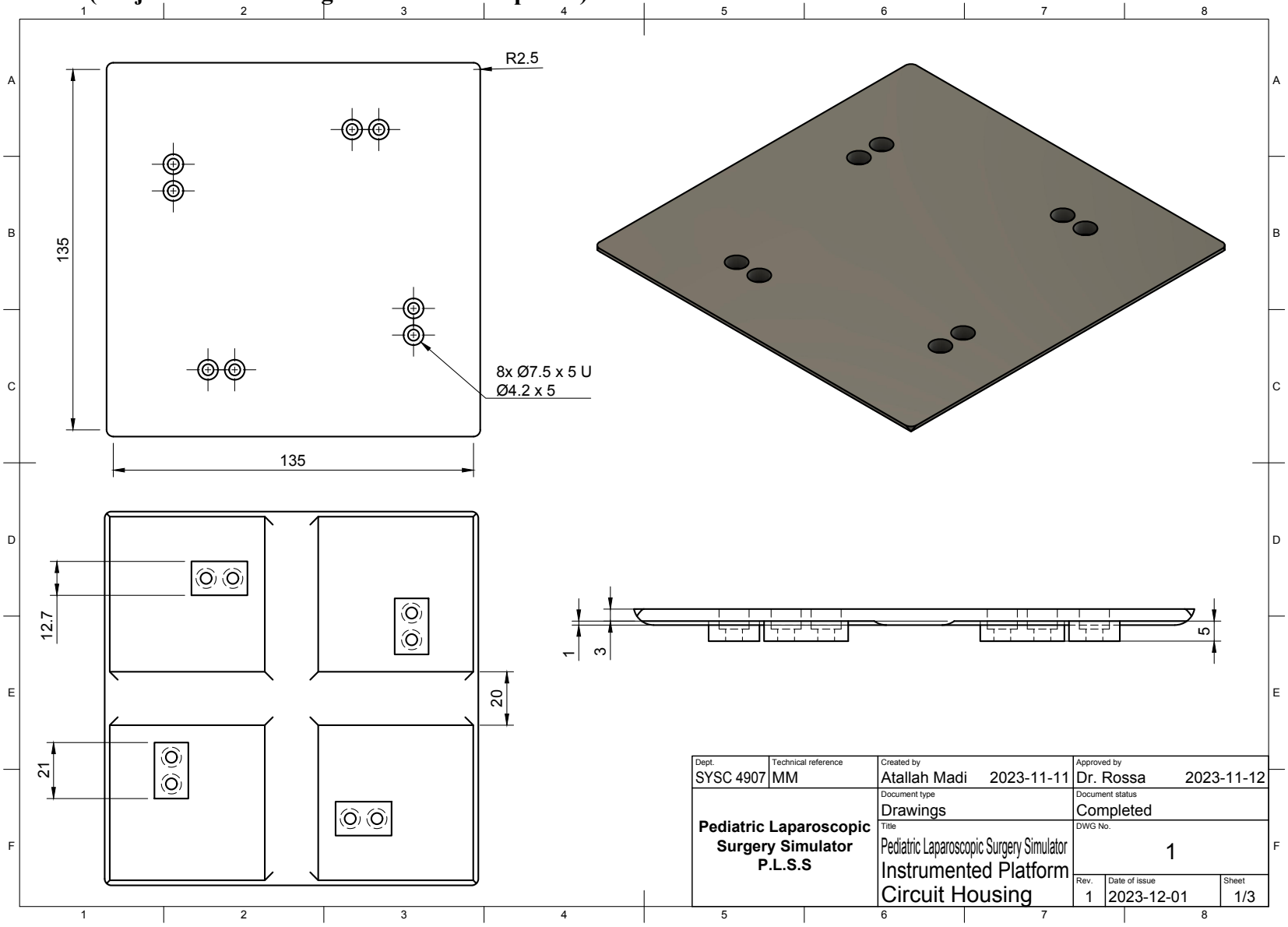
[30] TensorFlow 2 Detection Model Zoo. Available at: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

[31] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

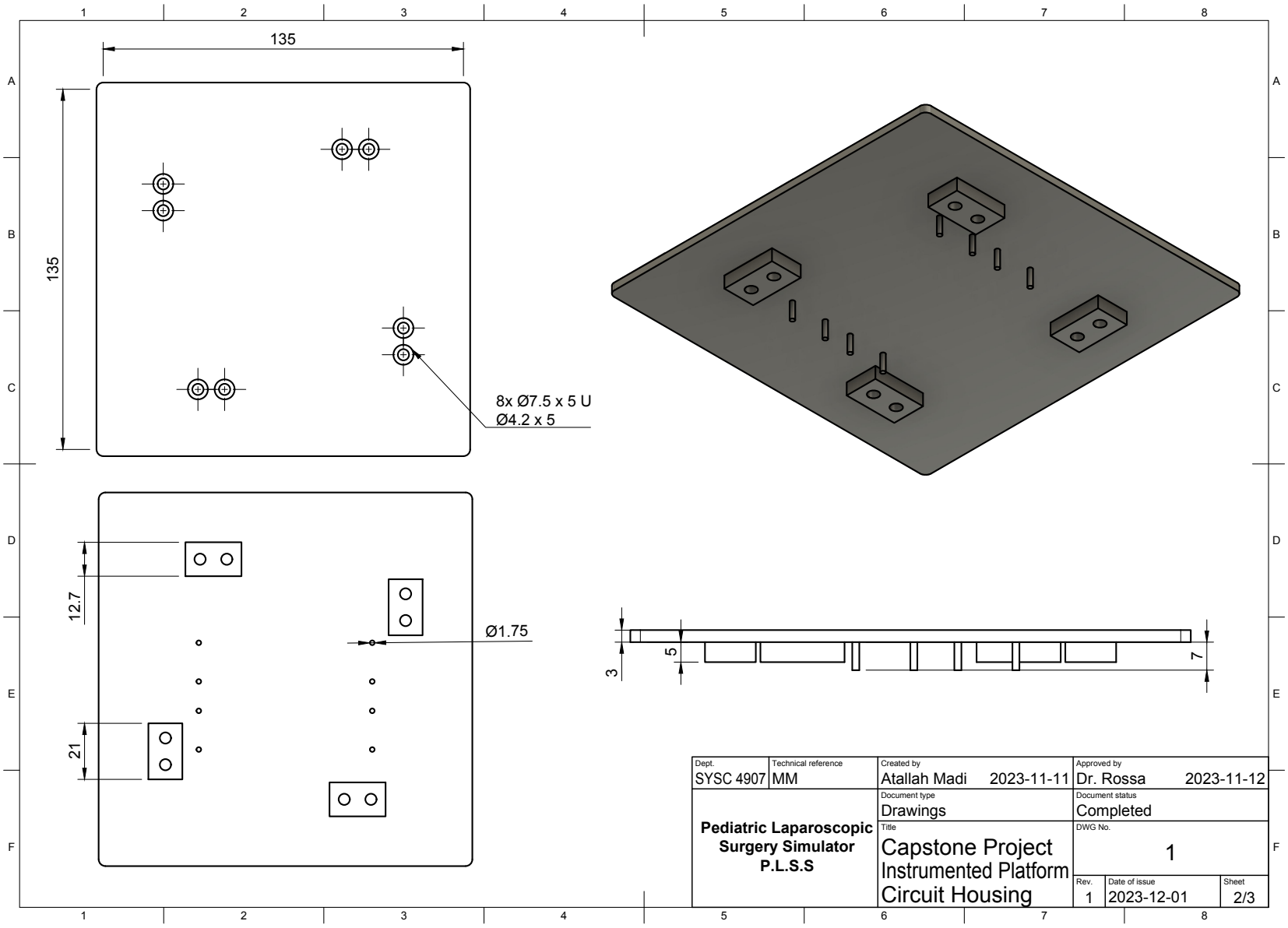
Appendix A (Project Schematic)



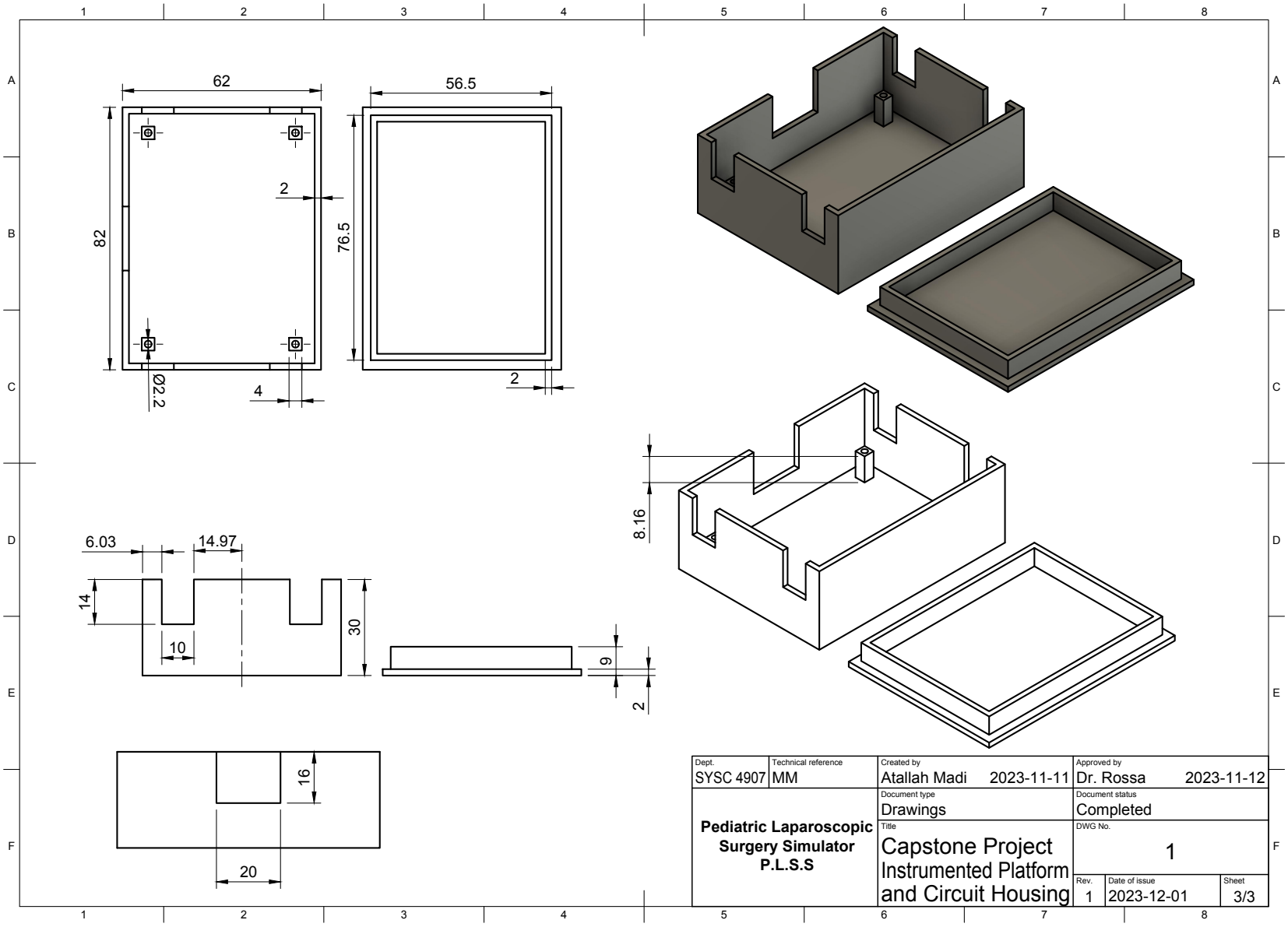
Appendix B (Project CAD Drawings: Force Plate Top Side)



Appendix B (Project CAD Drawings: Force Plate Bottom Side)



Appendix B (Project CAD Drawings: Circuit Housing)



Appendix C (Project UML Diagram)