DRL-based low Latency control for Endoscopic operations

Abdelrahman Soliman*, Hoda Helmy*, Elias Yaacoub*, Mohamed Mabrok**,
Nikhil V. Navkar‡, and Amr Mohamed*
*Department of Computer Science and Engineering, Qatar University, Qatar
**Department of Mathematics, Statistics, and Physics, Qatar University, Qatar
‡Department of Surgery, Hamad Medical Corporation, Qatar
Email: as1701600@qu.edu.qa, hoda.helmy@qu.edu.qa, {amrm, eliasy}@ieee.org,
m.a.mabrok@qu.edu.qa, nnavkar@hamad.qa

***Abstract*— Minimally invasive surgery has seen significant advancements with the introduction of robotic systems, which are highly desirable due to their ability to enhance treatment scalability and precision. This study aims to develop an effective and intelligent system for controlling and streaming the endoscope camera during endoscopic operations. The proposed system leverages head motion data from the HoloLens inertial measurement unit (IMU) to control the endoscope camera's robotic arm. Additionally, a Deep Reinforcement Learning (DRL) technique is employed to manage the region of interest adaptively (ROI), thereby mitigating wireless channel impairments in the operating room and improving the surgeon's interaction and quality of experience (QoE).**

We developed a proof-of-concept by interfacing the HoloLens with the Gazebo simulation environment for robotic arm control. The DRL model demonstrated its efficacy by intelligently reducing the communication delay and enhancing image quality by incorporating machine learning techniques. Specifically, the DRL model reduced delay by 12.56% and increased image quality by 26.6%. Furthermore, a fixed frame technique resulted in an additional delay reduction of 18%.

The successful establishment of the proof-of-concept and the comprehensive analysis of the findings underscore the potential impact of our contribution to advancing intelligent, efficient, and easily controllable endoscopic surgical procedures. Our study highlights the importance of integrating advanced machine learning technologies and augmented reality interfaces to enhance the performance and reliability of robotic surgery systems, ultimately benefiting both surgeons and patients through improved surgical outcomes and experiences.

***Index Terms*— Deep reinforcement learning, Assistive Control, Augmented reality, minimally invasive surgery, endoscopy, Simulation.**

I. Introduction

Endoscopy is a medical imaging technique that allows doctors to examine internal organs and cavities using an endoscope. Initially, it was primarily utilized to diagnose gastrointestinal issues, but subsequently, it has been adopted for performing minimally invasive surgeries (MIS). A flexible endoscope is a long, thin, flexible tube with a light and a camera attached to its tip that can be inserted inside a patient's body to access the internal organs [1]. The surgeon can visually examine a concerned area by obtaining images on a video monitor.

During an endoscopy procedure, it is essential to have an assistant who can effectively manage and position the endoscope as per the surgeon's directions. The assistant's involvement plays a vital role in the procedure's success. Nonetheless, miscommunication between the assistant and the surgeon can cause challenges and potentially influence the procedure's results [2]. Integrating robot arms into endoscope operations enhances precision and control, allowing surgeons to manipulate the endoscope camera directly. This transfer of control can be further streamlined through augmented reality (AR) technology, which can provide real-time, intuitive guidance, thus simplifying the camera manipulation process and reducing the dependency on assistants.

The medical sector has recently widely adopted virtual and augmented reality devices for surgical training. However, the benefits of augmented reality devices go beyond training and can extend to other areas, such as video endoscopy monitoring. The Inertial Measurement Unit (IMU) of a head-mounted device, for example, can be leveraged to provide better control over the position of the endoscope [3] [4].

To enhance the surgeon's experience, we directed our efforts toward improving the quality of the video feed to provide a more satisfactory real-time experience. The motivation for seeking a balance between video quality and delay stems from challenges inherent in wireless communication channels, especially in operating rooms where numerous electronic devices can cause interference. Despite dedicated connections, wireless channels are susceptible to fluctuating conditions like multipath effects and electromagnetic interference, potentially leading to increased delays and packet loss. Thus, optimizing the video frame quality while managing delay becomes crucial to ensuring real-time transmission and maintaining high-quality imaging during endoscopic procedures.

Our proposal involves using an adaptive Region of Interest (ROI) detection system and an intelligent Deep Reinforcement Learning (DRL) model to balance the overall quality of the video frame with the amount of delay it incurs.

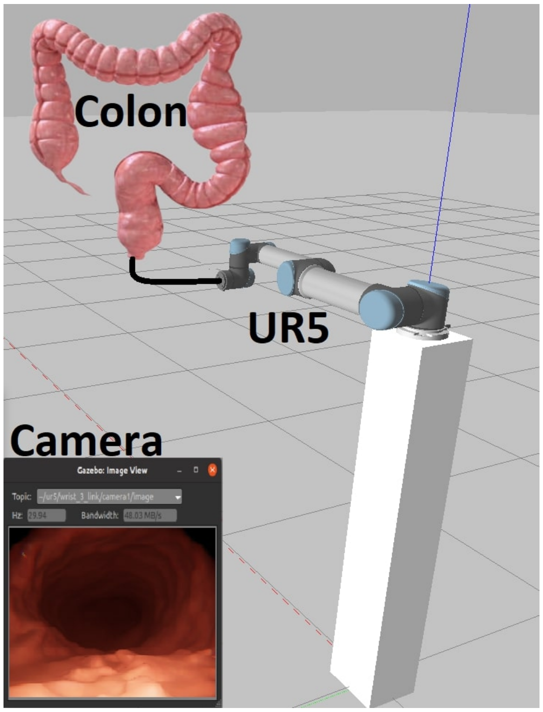


Fig. 1: Gazebo simulation and navigation through human body

We employ a higher-quality ROI than the background (non-ROI) and dynamically adjust ROI size based on network circumstances. The surgeon can make educated selections with high-quality ROI displays. With the suggested approach, its size will vary dynamically based on network conditions. The non-ROI zone can be presented at reduced quality, requiring lower network data rates. Higher data rates for higher ROI quality cause longer transmission delays, especially on networks with low capacity. Thus, DRL chooses the smallest ROI size and lowest non-ROI quality when network circumstances are poor and the converse when they are excellent. This method reduces endoscopic delays and improves video transmission latency without affecting ROI.

The contributions of this paper can be summarized as follows:

- We provide an enhanced adaptive ROI DRL solution for more responsiveness and low latency in the wireless stream for the HoloLens.
- We develop a proof-of-concept for the whole control system, including the simulation environment for the endoscope robotic arm. And the interfacing for the HoloLens head motion control.
- Our proposal extends to an integrated approach, seamlessly combining the DRL solution with the HoloLens head motion control to achieve a cohesive and efficient system.

The structure of this paper is as follows: Section II provides an overview of related work; Section III outlines the system architecture, and Section IV describes how the system was constructed in the simulation environment, along with the DRL approach. Section V discusses the DRL results in this work and showcases the simulation results. Finally, Section VI concludes the paper and outlines future research directions.

A. Related Work

In this section, we will explore the research and the initial steps conducted by other scholars in the field. Firstly, We will dig into the application of Artificial Intelligence in telemedicine, focusing on its role in minimizing latency and enhancing performance. Subsequently, we will discuss the various approaches for controlling surgical robots, emphasizing endoscopic robots. Furthermore, we will explore the contemporary utilization of augmented reality in this domain to achieve improved outcomes.

Reducing the latency or delay of video streams has been a critical area of research in recent years, and many solutions have been suggested. These techniques include adaptive bit rate streaming, edge computing, and network optimization. Adaptive bit rate streaming has been a popular approach for reducing the delay of video streams. Work in [5] proposed an adaptive transmission technique for a delay-constrained wireless video that dynamically adjusts the transmission rate based on the buffer occupancy and the delay requirements of the video. For video encoding, they also use adaptive slice partitioning and intra-refreshment. Simulations and experiments are used to test the proposed method, which shows that it can get better video quality with less delay than existing methods. Similarly, [6] introduced an adaptive transmission rate control algorithm that models the problem of video rate control over a time-varying channel as a Markov decision process (MDP). They generated two algorithms, one based on a greedy approach and the other on a stochastic gradient descent (SGD) method. These publications are similar to ours because our DRL model seeks to adaptively choose ROI size and non-ROI quality based on network conditions. However, these methods only take action depending on the current conditions, while our intelligent solution takes action depending on the expected throughput after analyzing the history of network traces.

Edge computing has also been explored as a solution to lower the latency of video streams. Work in [7], [8] proposed an edge computing-based framework that offloads the processing tasks to edge servers, reducing the latency and improving the user experience. They found that their proposed framework could significantly improve system performance compared to the standard Dynamic Adaptive Streaming over HTTP (DASH)-based system with fixed bitrates for transcoding. This is because their proposed framework's method for transcoding is more efficient, and the bandwidth adjustment can make better use of the bandwidth. Additionally, authors in [9] presented a model that utilizes edge computing to enhance video quality and reduce the delay in video transmission. They also concentrated on the possibilities and effects of cutting-edge technologies on multimedia. The authors concluded that cutting-edge technologies could help solve many problems with interactive media and video streaming.

Furthermore, researchers have explored other techniques to reduce the delay of video streams. One example is a suggested intelligent algorithm for wireless networks that uses adaptive quantization to change the quantization levels of video frames based on the packet loss rate and the quality of experience (QoE) needs [10]. They also implemented an application model framework that automatically assessed QoE. The model detects network packet loss and chooses the best QP value to improve end-user QoE. Based on our research and related work, no intelligent, practical solution has been investigated for combining machine learning and network condition tracking to reduce video stream delay, especially in healthcare.

Researchers have recently explored new ways to control endoscope cameras without a human operator. This research started with proposals for voice commands in 1998 [11], postural control in 2009 [12], and gaze-contingent control in 2010 [13]. Single-hand control is another method of control that [14] and [15] have proposed. The first system to use a head-mounted display device (HMD) was developed in 2010 [16]. It allows surgeons to control the endoscope camera using head movements by adding a sensor that reads the orientation of the surgeon's head, which frees up their hands for instrument operation. The system uses a flexible endoscope that offers more space for instrument maneuvering and can display the endoscopic image on a head-mounted display. They could achieve the pitch and yaw movements solely, and the image display in the HMD required a zooming function for better view focus and image.

Surgeons have increasingly opted for AR and VR technologies in robotic surgeries before expanding research to use AR and VR devices to conduct remote surgeries. These technologies allow surgeons to view complex anatomical structures and surgical procedures in real time, providing an immersive experience that can improve precision and accuracy. In [17], they utilized simulators based on VR technology to enable trainees to interact with virtual organs and tissues using the same surgical tool handles used in actual minimally invasive surgery (MIS). During the training, trainees could view images of tool-tissue interactions on a monitor, providing a realistic experience that closely imitated actual laparoscopic procedures. VR simulators allowed the trainees to have a tactile and interactive experience, which could enhance their skills and proficiency in a safe and controlled environment. Similarly, [18] combined an AI module with AR technology to create trajectories for surgical procedures. The AI module can use reinforcement learning to learn how to perform operations and produce trajectories according to the current task. To make the learning process more intuitive and immersive, the trajectory plan is projected onto a stereo video, which provides trainees with intelligent guidelines in 3D AR.

In [19], the authors presented a new method for positioning an endoscope camera during minimally invasive surgery. The endoscope holder, often a robotic or actuated mechatronic system, can adjust its position automatically based on the surgeon's focus, as detected by a virtual reality headset, and the movement of surgical instruments, analyzed in real-time using a neural network called YOLOv3 and ResNet. To properly center the camera during surgery, the robot must identify the surgeon's region of interest (RoA), typically defined around the surgical instruments.

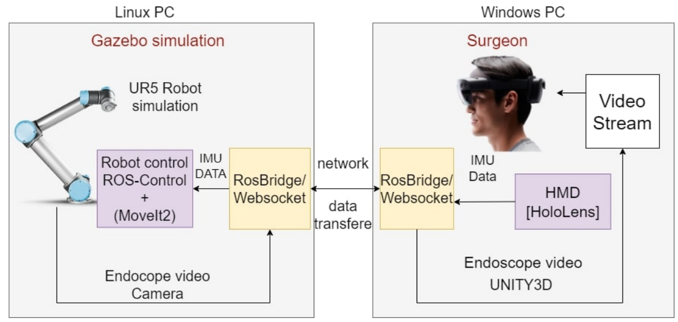


Fig. 2: System Model Overview.

Recently, in [20], the authors proposed two approaches for controlling augmented reality by visualizing robotic stereo flexible endoscopes with the help of HoloLens. Markers on the patient-side manipulators are tracked by the HoloLens to establish a link between the surgeon's head and the base frame of the augmented reality for viewing robotic stereo flexible endoscopes. The control system offers two primary modes: hybrid visual servoing control mode, which tracks the surgical instrument and directs the endoscope toward the specific area of interest, and head tracking control mode, where the position of the surgeon's head is tracked to adjust the observation angle accordingly. The surgeon can switch between these modes using foot pedals.

While previous methods significantly reduce the need for manual adjustments by human assistants, they do not entirely eliminate it. Our project builds on this work by using two-way wireless communication, leveraging the Hololens for IMU data to move the endoscope camera, and streaming the endoscope video directly to the head-mounted display (HMD). The key difference is our integration of intelligent machine learning algorithms and network status tracking to minimize video stream delay, a novel contribution not addressed in previous research.

II. SYSTEM ARCHITECTURE

A. System model

The endoscopy field presents surgeons with many challenges, including efficiently controlling the endoscope camera without needing an in-room assistant. To address this challenge, we have developed an innovative system utilizing augmented reality devices, empowering surgeons to manipulate the view of the endoscope camera according to their desired perspective. Another significant hurdle is the impact of wireless channel variability in the operating room, which can lead to data transmission delays and degrade the quality of the endoscope video stream. To tackle this issue, we introduce a deep reinforcement learning (DRL) based model that intelligently reduces delay. This model dynamically adjusts the size of the Region of Interest (ROI) in the video frame to be displayed at the highest quality while optimizing the quality of the non-ROI region based on the anticipated throughput. By tackling these obstacles, our research aims to enhance the precision and efficiency of endoscopy procedures, fostering improved surgical outcomes and patient care.

Our system architecture comprises multiple integral components, including the HoloLens, a robotic holder that supports the endoscope camera in a simulated environment, a computer processing unit, and specialized software such as Unity3D, GAZEBO (running on ROS2), RVIZ, and Moveit libraries. During the endoscopic procedure, the physician wears the head-mounted display (HMD) or HoloLens to visualize augmented images and provides input through the integrated inertial measurement unit (IMU) to navigate and manipulate the endoscope.

The computer processing unit assumes a critical role in the system, as it receives images obtained from the endoscope camera and facilitates their transmission to the HoloLens. Simultaneously, it tracks the head movements of the physician, ensuring synchronized visual feedback. Through intuitive head tilting, the physician can adjust the endoscope's orientation, enabling effortless navigation within the patient's body and augmenting maneuverability throughout the procedure.

In Figure 2, the system architecture illustrates integrating two distinct operating systems (OS) on separate devices, essential for establishing communication between remote systems via the ROS bridge. This model incorporates two simulation engines, GAZEBO and Unity3D, which are pivotal in the system's development and functionality. The decision to utilize separate machines rather than alternative approaches like Windows Subsystem for Linux (WSL) or a Linux virtual machine is justified by the need for dedicated hardware resources and real-time performance that these simulations demand. By distributing tasks across distinct OS environments, our approach optimizes system responsiveness and effectively mitigates latency in endoscopic video streaming.

B. Virtual Simulation

This section will discuss each contribution within our system architecture, beginning with constructing the virtual simulation environment. We will then discuss the configuration of the Hololens device with Unity, followed by the data transfer mechanisms between the operating systems. Finally, we will explore the application of the DRL technique to enhance video streaming performance under varying wireless channel conditions.

1) GAZEBO and the simulated world: Creating a virtual simulation allows us to test and adjust our methods securely. Our architecture uses the Universal Robot 5 (UR5), a flexible robotic arm with six degrees of freedom. The MoveIt2-configured UR5 controller works flawlessly with an open-source URDF file. This setup allows precise robot motion planning, control, and manipulation. Task-specific end effectors can be used with the UR5. We attach a camera to the UR5's robot arm tip for our project.

We use Gazebo11, a robust 3D open-source simulator, to mimic the UR5 robot. Gazebo11 lets us use the robot's physical properties to create realistic scenarios in a virtual environment. Gazebo11's plugins allow researchers to test the robot's capabilities in a secure simulated environment before applying them to a real robot.

Gazebo11 is used with ROS2, a middleware for robot application development, for easy integration and optimization. Our project used ROS2 Foxy on Ubuntu Linux - Focal Fossa (20.4). The Unified Robot Description Format (URDF) was used to model the robot's linkages, joints, movements, attributes, and sensors. Additionally, MoveIt2, a full motion planning framework, offers several tools and libraries for kinematics, collision detection, trajectory execution, and modification. MoveIt2 simplifies complex robot application development in our study with its user-friendly high-level APIs and intuitive UI.

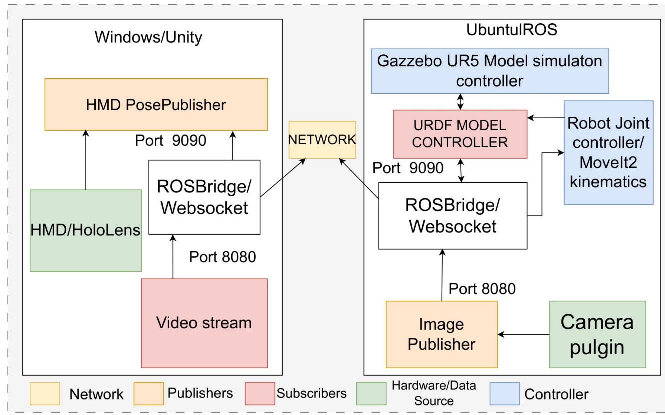


Fig. 3: Data flow in the system.

2) Unity3D and ROSBridge: Meanwhile, the surgeon's head movement controls the UR5 robot. The surgeon wears a Unity3D-configured HMD. Unity3D lets the surgeon view the endoscopic camera's output in the Gazebo model without interruption. Unity3D uses HMD IMU data to position the robotic arm. With this information, the endoscopic camera can be adjusted to the surgeon's chosen view without an assistant.

ROS Bridge connects simulation components. This bridge refers to packages that enable bidirectional system communication. It streamlines ROS1 and ROS2 framework interface and operating system communication. Our experiment used the ROS Bridge to send HMD IMU data to the UR5 robot and UR5 camera data to the HMD display. These components can exchange data easily with this integration.

As shown in Figure 3, the UR5 robot's virtual simulation is integrated using ROS2 and the Gazebo simulation environment. This simulation setup is Linux-only. However, Unity3D on a separate Windows system configures and integrates the HMD. The HMD uses IMU data to track the surgeon's head motion, mostly orientation. Unity3D, which processes head motion data, collects IMU data. Through ROS Bridge messages, IMU data is published as a pose publisher. ROS Bridge uses TCP/IP to connect Linux and Windows workstations.

For efficient data transfer, our system has two data channels: one for IMU data and one for endoscopic camera streaming. Ubuntu receives IMU data from the ROS Bridge. ROS inverse kinematics and MoveIt2 libraries calculate and control the robotic arm's 6 DOF utilizing orientation posture data. The goal is to reposition the robotic arm's end effector to match the surgeon's perspective of the targeted area. Inverse kinematics modifies the 6 DOF robotic arm's motion. The joint controllers use the MoveIt2 library to change the UR5 model's Unified Robot Description Format (URDF) in the simulated world to allow motion. Simulations use a plugin camera as the endoscopic camera. The plugin camera in the simulated Gazebo world streams live footage to Unity3D via an image publisher to the ROS Bridge.

C. Deep Reinforcement Learning

After defining the system model and presenting the simulation architecture and the data flow, we will discuss an intelligent approach to reduce the stream delay. Leveraging machine learning approaches to reduce delay is significant because it helps us better comprehend the complexities and hidden correlations between data and outcomes. Our preference for employing DRL stems from the requirement for a sophisticated decision-making model that considers the previous, current, and following possible states; one could be the throughput. DRL comprises several constituent elements that will be elaborated upon in the subsequent subsections.

1) Environment Design: The environment is where we carry out actions and obtain states or observations. In our case, the environment consists of the video frames captured by the camera on our simulated robot. As shown in Figure 4, the first step is to grab the frame by subscribing to the endoscope camera topic. Then, the agent performs actions depending on the value of the Boolean skip frame. If true, it will skip the original ROI detection and use the previous actions. When false, it will use the customized shallow convolutional neural network (S-CNN) to get the original ROI and then apply the new actions. After getting the ROI and the actions, we start compressing the non-ROI region and merging it with the ROI to get the final frame. To evaluate the actions taken, we compute the observations and reward. Finally, the final frame is published to a new topic that Unity will use for streaming to the HoloLens.

2) Action Spaces and States: Action space consists of three continuous parameters, each of which is normalized between zero and one to assist the model in enhancing its training while also avoiding divergence [21]. The action space is as follows: A=$\{X, Y, QF\}$, where QF is the quality factor, and X and Y are the measures of how much the original width and length of the ROI should increase.

Each state (s) in the state set S consists of delay, quality, and throughput. Hence, s=$\{\gamma, \lambda, T\}$, where (γ) is the total delay, which is the sum of the communication delay and the processing delay. It is given by:

$$\gamma = (S(x, y, q)/throughput) + (ProcessingDelay) \quad (1)$$

(λ) is the quality, and (T) is the throughput. Equation (1) is used to determine the communication delay. In the numerator, we calculate the total size of the frame in bytes using the regression model shown in Equation (2),

$$S(x, y, q) = p_{00} + p_{10} \cdot x + p_{01} \cdot y + p_{20} \cdot x^2 + p_{11} \cdot x \cdot y + p_{02} \cdot y^2 + p_{30} \cdot x^3 + p_{21} \cdot x^2 \cdot y + p_{12} \cdot x \cdot y^2 + p_{03} \cdot y^3 \quad (2)$$

In (2), the following are the values for the regression model's parameters that were obtained using the MATLAB regression tool with an R-Square of 0.822:
$p_{00} = 6.256 \times 10^4$, $p_{10} = -0.2356$, $p_{01} = 432.4$, $p_{20} = 1.412 \times 10^{-6}$, $p_{11} = 0.001398$, $p_{02} = -8.561$, $p_{30} = -2.637 \times 10^{-12}$, $p_{21} = -8.87 \times 10^{-9}$, $p_{12} = 6.147 \times 10^{-6}$, $p_{03} = 0.04034$.

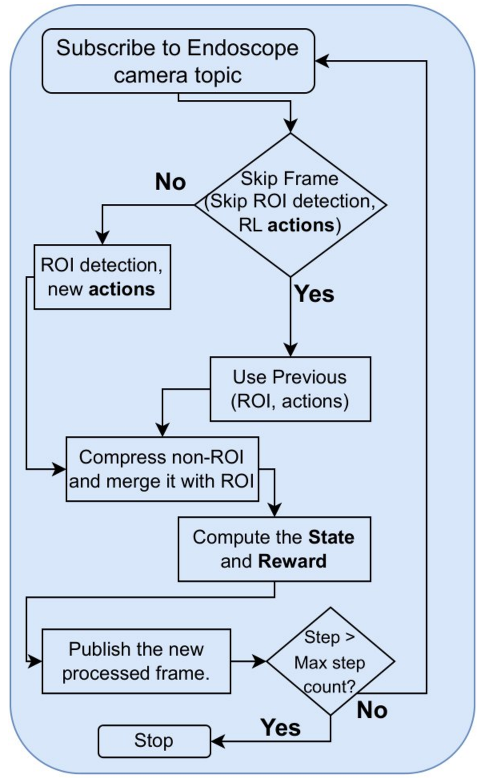


Fig. 4: RL flowchart.

The original ROI width + action (X), original ROI height + action (Y), and quality factor action (QF) are the inputs for the regression model. Then, to obtain the transmission delay, we divide the frame size by the throughput in the first part of Equation (1). The throughput is highly variable and dynamic, especially in surgery rooms, where reliable and efficient communication is critical to the success of procedures and where changes in throughput can significantly affect network performance. However, throughput can vary depending on network congestion, interference, distance, and network setup. Physical barriers such as walls, floors, and medical equipment can block or weaken the wireless signal, leading to interference and decreased network performance. Other wireless devices in the surgery room, such as monitoring equipment, can also interfere with the wireless signal and cause slow or dropped connections, further exacerbating the variability in throughput. Additionally, high volumes of network traffic, especially during critical surgical procedures, can cause congestion and slow down the network, leading to more variability in throughput. The quality and capabilities of the network equipment, including access points and antennas, must be of high quality to guarantee stable and efficient connections over the wireless network's frequency band and minimize variability in throughput. We measured the network traces between the client and server, which are the HoloLens and the stream computer, using the Ookla speed test framework [22]. This test was run multiple times at various periods of the day to simulate different network congestion scenarios.

3) Reward Function: The reward function in equation (3) demonstrates how it directs and controls the model's actions to help it achieve specific objectives. Any deviation from the stated goals of our reward function, i.e., minimizing delay and maximizing quality, is considered a negative reward, i.e., a penalty. In our setup, we discovered that throughput values less than 103,076 MB/s result in unacceptable delays. So, we have used this value as a throughput reference point to inform the model that the throughput below this point is poor. Additionally, we added that a negative reward should be given if the sum of the actions is below or equal to 0.2 when throughput is low and greater than or equal to 0.9 when throughput is high.

$$\textbf{Reward}(\gamma, \lambda, T, A) := \begin{cases} (\gamma^{-1} + \lambda^{-1}) & \text{if } T < 103,076 \,\&\, \Sigma A \leq 0.2 \\ (\gamma + \lambda) & \text{if } T \geq 103,076 \,\&\, \Sigma A \geq 0.9 \\ (-1) & \textit{else} \,. \end{cases} \tag{3}$$

4) Deep Reinforcement Learning Algorithm: The agent aims to find the policy that maximizes the overall discounted reward. Since basic RL algorithms cannot handle continuous actions, we can utilize a Deep Deterministic Policy Gradient (DDPG) or Soft Actor-Critic (SAC). The SAC algorithm was chosen because it was stable in our context and produced superior results. A further factor is that authors in [23] concluded that even with some continuous action spaces that are not particularly complicated, DDPG works well. Still, it does not have a rapid convergence rate. It is well known that SAC can train and test a stochastic policy using entropy regularization and in an off-policy manner. Moreover, it uses a separate network for policy and one for value functions, following an actor-critic design. Generally, SAC algorithm as shown in Algorithm 1, aims to learn the policy $\pi_\phi(a_t|s_t)$, Soft Q-value $Q_\theta(s_t, a_t)$ and soft state value $V_\psi(s_t)$. The Soft Q-value function (7) is obtained from the Bellman equation (4) and soft state value (5).

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \rho_\pi(s)}[V(s_{t+1})] \tag{4}$$

$$V(s_t) = \mathbb{E}_{a_t \sim \pi}[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t)] \tag{5}$$

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{(s_{t+1}, a_{t+1}) \sim \rho_\pi}[Q(s_{t+1}, a_{t+1}) \tag{6}$$

$$-\alpha \log \pi(a_{t+1}|s_{t+1})] \tag{7}$$

The parameters of Q-function and the policy networks are ψ, θ, ϕ, and $\bar{\psi}$; ϵ_t is an input noise vector. The gradient in Equation (8) trains the soft value function to minimize the squared residual error.

$$\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(s_t) \left(V_\psi(s_t) - Q_\theta(s_t, a_t) + \log \pi_\phi(a_t|s_t) \right) \tag{8}$$

Also, using stochastic gradients as shown in Equation (9), The soft Bellman residual can be reduced and improved upon by training the soft Q-function parameters.

$$\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(a_t, s_t) \left(Q_\theta(s_t, a_t) - r(s_t, a_t) - \gamma V_{\bar{\psi}}(\mathbf{s}_{t+1}) \right) . \tag{9}$$

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $a_t \sim \pi_\phi(a_t|s_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|s_t, a_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  end for
end for

III. Results and Discussion

This section presents a detailed analysis of the training results of the Deep Reinforcement Learning (DRL) model. We also investigate the impact of model implementation on latency and quality, highlighting the new changes and improvements that have been achieved. Furthermore, we compare our suggested solution's performance with the baseline and benchmark solutions. Lastly, we propose a novel solution to reduce processing time.

A. S-CNN Results

The S-CNN model was trained using Python and the TensorFlow library, a widely adopted deep learning framework. The Adam optimizer, known for its robustness and fast convergence rate, was employed with default settings during the training phase. To facilitate the training process, an endoscope frame dataset [24] was utilized. This dataset consists of 1540 frames and an expert in the field of endoscopy manually labelled each frame's ROI. To augment the data, various image augmentation techniques, such as crop, rotation, and filtering, were utilized to enhance the variability of the data while preventing over-fitting. The final dataset comprised over 30,000 frames, significantly more than the original sample size. The evaluation of the model's performance, as depicted in Figure 5, indicates that both the train and test accuracy scores are notably high. Consequently, the model exhibits a remarkable ability to identify the ROI accurately.

B. DRL training Convergence

After 15,000 timesteps of training at a learning rate of 0.002, as shown in Figure 6, the average episode reward converged to around 56. Figure 7 shows how DRL performs when the throughput is low and high. The green rectangle is the original ROI detected using the S-CNN model, while the purple rectangle is the action of the DRL, depending on

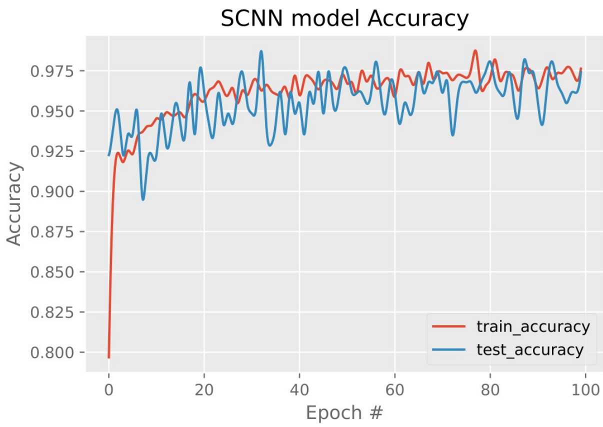


Fig. 5: SCNN model accuracy

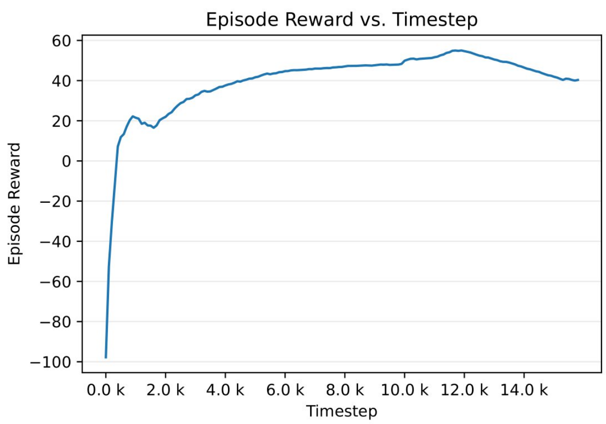


Fig. 6: Episode convergence for DRL training

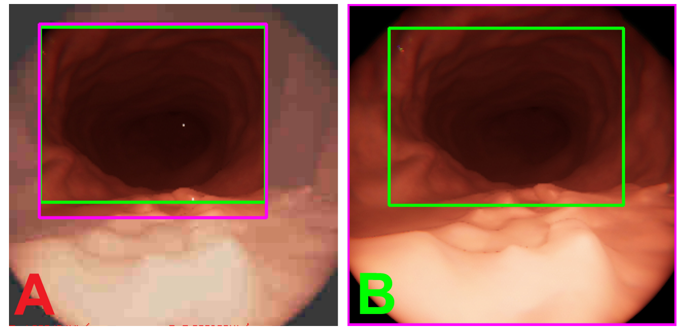


Fig. 7: A. ROI size when throughput is low, B. ROI size when throughput is high

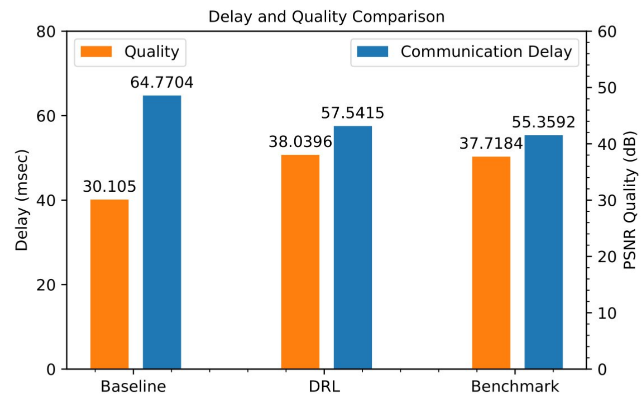


Fig. 8: Comparative study with the Baseline and Benchmark

the state. Also, it is notable that the quality of the non-ROI region of the image (A) is degraded due to the low throughput condition. When network conditions allow, the whole picture is rendered highly, as seen in image (B), where the purple rectangle engulfs the entire image.

C. Comparative study with the Baseline and Benchmark

Comparing the DRL technique to a baseline and Benchmark is necessary to assess its performance. The Baseline scenario has the lowest ROI and non-ROI quality, while the Benchmark scenario has the best possible ROI size and non-ROI quality. The image quality is measured using the peak signal-to-noise ratio (PSNR) metric. As shown in Figure 8, experiment results show that the DRL algorithm significantly improves PSNR quality, with a score of approximately 26.6% higher than the baseline (38.04 compared to 30.11). Also, the communication delay is close to the benchmark value.

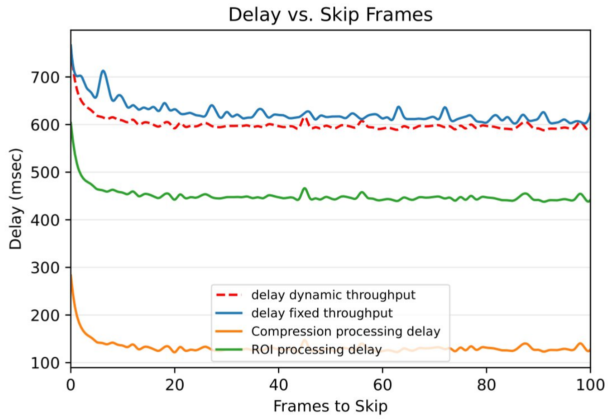
Overall, the DRL algorithm offers promising results for both total delay and PSNR quality, with a significant improvement of 26.6% in PSNR quality over the baseline while decreasing the communication delay by 12.56%. These findings suggest that the DRL algorithm may be suitable for reducing delay while improving image quality.

D. Optimizing processing time

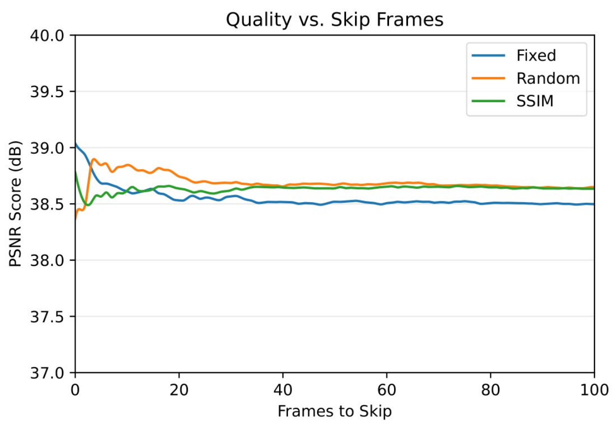
Although we assume in this work that we have high and powerful computational resources and focus mainly on the reduction of communication delay, we tried to further optimize the processing time by introducing an efficient technique that is necessary to maximize video processing for low-cost and low-latency streaming to minimize the recurrent prediction of the region of interest (ROI). Consequently, skipping these repetitive, time-consuming, and processing-intensive actions has been proposed. It has been demonstrated that skipping the prediction of ROI for some consecutive frames is acceptable due to the slow changes in the frames' characteristics and their substantial similarity. To confirm this, the Structural Similarity Index (SSIM) was applied to thirty consecutive frames, resulting in a high average SSIM score of 0.951, indicating that these frames are highly correlated. Furthermore, Figure 9a illustrates that skipping frames can significantly reduce the total delay, including communication and processing delays. Additionally, Figure 9b shows that skipping the frames does not significantly impact the quality.

To determine the optimal number of frames to skip, a fixed number is selected in each episode, and the average total delay is calculated. Additionally, the delay is compared in the scenarios where the throughput is fixed and dynamic to evaluate if the throughput significantly affects the total delay.

The results show that both approaches follow the same trend, and fixing the frames reduces the total delay. The



(a) Delay vs. Frames to skip



(b) Quality vs. Frames to skip

Fig. 9: Delay and Quality vs Frames to skip

results also show that the transmission delay, dependent on the throughput, does not dominate the total delay. Initially, the total delay decreased sharply, then stabilized after reaching 20 frames to skip. Thus, the optimal number of frames to skip is between 20 and 40.

Figure 10 presents multiple techniques for selecting the number of frames to skip, including a fixed number, random selection, and a dynamic approach using SSIM scores. Although dynamic techniques are more adaptable, comparing frames and computing SSIM requires more resources. However, the surgeon or user has the option to select this method of operation. The user could choose a dynamic technique that adapts to the current environment but at the cost of increased delay. The comparison among the three techniques for skipping frames, namely random, fixed, and SSIM difference, showed that all three techniques produced close quality scores, with fixed frame skipping outperforming the other two in total delay. Specifically, the total delay was 619.30 ms and 671.88 ms for the random and SSIM difference techniques, respectively. In comparison, the fixed technique reduced the total delay to 596.59 ms, a reduction of around 3.7% and 11.2%, respectively. In terms of quality, all three techniques produced similar results, with a quality score of around 38.5 dB to 38.7 dB.

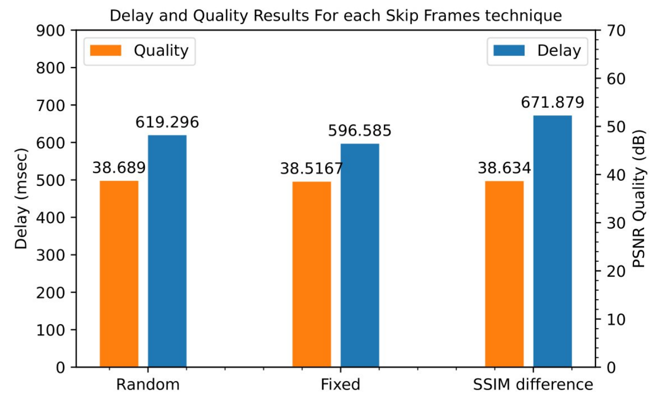


Fig. 10: Delay and quality comparison between skip frames techniques

Therefore, while all three techniques produced comparable quality, the fixed technique is the most efficient option for reducing processing time and achieving low latency in video processing. By leveraging this technique in the system, the average total delay was decreased by approximately 18%.

E. Simulation and Development results

In this subsection, we show the simulation results and the implementation of the control and streaming part. The head-controlled UR5 robot arm works as the endoscope that navigates inside the 3D human colon. Also, the camera stream is shown in the corner, which will later go to the DRL model and output the processed frame back to the HMD. [1]

IV. Conclusion and Future Work

This paper proposes a novel system and techniques integrating robotics, virtual reality, and machine learning technologies for controlling and streaming endoscopic operations. It was demonstrated that head motion from HoloLens IMU data could prevent the endoscope camera's robotic arm. Also, the DRL model can reduce wireless video stream latency using the adaptive ROI technique. The DRL model presented in this study reduced delay by 12.56% while increasing quality by 26.6%. Also, after using the fixed frames technique, the delay was reduced by 18%. We can conclude that the fixed number skip frame technique was the most effective method for lowering video processing time and achieving low latency. These contributions and findings substantially impact the development of intelligent, efficient, and readily controlled endoscopic surgical operations.

We can consider adding predefined speech commands to control the camera or the quality of future works and directions. Also, an automatic discovery and navigation option for surgeons will be helpful and vital for them to reach and detect some common disease markers. Finally, exploring the expansion of this work in other medical procedures such as

[1] A demo video can be accessed at https://youtu.be/E-_7eFx1ekA.

Esophagectomy, Laparoscopic Surgery, and da Vinci Surgical Robotic-Assisted Surgery.

Acknowledgements

This work was supported by an NPRP award (NPRP13S-0205-200265) from the Qatar National Research Fund (a member of The Qatar Foundation). The work of Abdelrahman Soliman was also supported by the GSRA award (GSRA10-L-2-0604-23091) from the Qatar National Research Fund (a member of The Qatar Foundation. The findings achieved herein are solely the responsibility of the authors.

References

[1] T. A. Visconti, J. P. Otoch, and E. L. Artifon, "Robotic endoscopy. a review of the literature," *Acta Cirúrgica Brasileira*, vol. 35, no. 2, 2020.

[2] B. M. Gillespie, W. Chaboyer, and N. Fairweather, "Interruptions and miscommunications in surgery: An observational study," *AORN Journal*, vol. 95, no. 5, pp. 576–590, 2012.

[3] J. S. Kim, W. C. Park, and J. H. Lee, "Comparison of short-term outcomes of laparoscopic-assisted colon cancer surgery using a joystick-guided endoscope holder (soloassist ii) or a human assistant," *Annals of Coloproctology*, vol. 35, no. 4, p. 181–186, 2019.

[4] R. Randell, S. Honey, N. Alvarado, J. Greenhalgh, J. Hindmarsh, A. Pearman, D. Jayne, P. Gardner, A. Gill, A. Kotze, and D. Dowding, "Factors supporting and constraining the implementation of robot-assisted surgery: a realist interview study," *BMJ Open*, vol. 9, no. 6, 2019. [Online]. Available: https://bmjopen.bmj.com/content/9/6/e028635

[5] C. Gong and X. Wang, "Adaptive transmission for delay-constrained wireless video," *IEEE Transactions on Wireless Communications*, vol. 13, no. 1, pp. 49–61, 2014.

[6] R. Doreswamy, A. G. Colaco, V. Sevani, P. Patil, and H. Tyagi, "Rate adaptation for low latency real-time video streaming," in *2023 National Conference on Communications (NCC)*, 2023, pp. 1–6.

[7] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, "Adaptive wireless video streaming based on edge computing: Opportunities and approaches," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 685–697, 2019.

[8] S. Gül, D. Podborski, T. Buchholz, T. Schierl, and C. Hellge, "Low-latency cloud-based volumetric video streaming using head motion prediction," ser. NOSSDAV '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 27–33. [Online]. Available: https://doi.org/10.1145/3386290.3396933

[9] K. Bilal and A. Erbad, "Edge computing for interactive media and video streaming," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 68–73.

[10] M. Taha and A. Ali, "Smart algorithm in wireless networks for video streaming based on adaptive quantization," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 9, p. e7633, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.7633

[11] L. Mettler, M. Ibrahim, and W. Jonat, "One year of experience working with the aid of a robotic assistant (the voice-controlled optic holder aesop) in gynaecological endoscopic surgery," *Human Reproduction*, vol. 13, no. 10, p. 2748–2750, 1998.

[12] A. Minor Martinez, J. Villalobos Gomez, R. Ordorica Flores, and D. Lorias Espinoza, "Postural mechatronic assistant for laparoscopic solo surgery (pmass)," *Surgical Endoscopy*, vol. 23, no. 3, p. 663–667, 2009.

[13] D. P. Noonan, G. P. Mylonas, J. Shang, C. J. Payne, A. Darzi, and G.-Z. Yang, "Gaze contingent control for an articulated mechatronic laparoscope," in *2010 3rd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2010, pp. 759–764.

[14] E. D. Rozeboom, J. G. Ruiter, M. Franken, M. P. Schwartz, S. Stramigioli, and I. A. Broeders, "Single-handed controller reduces the workload of flexible endoscopy," *Journal of Robotic Surgery*, vol. 8, no. 4, p. 319–324, 2014.

[15] T. Iwasa, R. Nakadate, S. Onogi, Y. Okamoto, J. Arata, S. Oguri, H. Ogino, E. Ihara, K. Ohuchida, T. Akahoshi, and et al., "A new robotic-assisted flexible endoscope with single-hand control: Endoscopic submucosal dissection in the ex vivo porcine stomach," *Surgical Endoscopy*, vol. 32, no. 7, p. 3386–3392, 2018.

[16] R. Reilink, G. de Bruin, M. Franken, M. A. Mariani, S. Misra, and S. Stramigioli, "Endoscopic camera control by head movements for thoracic surgery," *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 510–515, 2010.

[17] C. Basdogan, M. Sedef, M. Harders, and S. Wesarg, "Vr-based simulators for training in minimally invasive surgery," *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 54–66, 2007.

[18] Y. Long, J. Cao, A. Deguet, R. H. Taylor, and Q. Dou, "Integrating artificial intelligence and augmented reality in robotic surgery: An initial dvrk study using a surgical education scenario," in *2022 International Symposium on Medical Robotics (ISMR)*, 2022, pp. 1–8.

[19] K. Zinchenko and K.-T. Song, "Autonomous endoscope robot positioning using instrument segmentation with virtual reality visualization," *IEEE Access*, vol. 9, pp. 72 614–72 623, 2021.

[20] X. Ma, C. Song, L. Qian, W. Liu, P. W. Chiu, and Z. Li, "Augmented reality-assisted autonomous view adjustment of a 6-dof robotic stereo flexible endoscope," *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 2, pp. 356–367, 2022.

[21] N. Rao, E. Aljalbout, A. Sauer, and S. Haddadin, "How to make deep rl work in practice," *ArXiv preprint arXiv:2010.13083*, 10 2020.

[22] S. by Ookla. (2006) Speedtest. [Online]. Available: https://www.speedtest.net/

[23] K. Ota, T. Oiki, D. K. Jha, T. Mariyama, and D. Nikovski, "Can increasing input dimensionality improve deep reinforcement learning?" *ArXiv preprint arXiv:2003.01629*, 2020.

[24] K. B. Ozyoruk, G. I. Gokceler, T. L. Bobrow, G. Coskun, K. Incetan, Y. Almalioglu, F. Mahmood, E. Curto, L. Perdigoto, M. Oliveira, H. Sahin, H. Araujo, H. Alexandrino, N. J. Durr, H. B. Gilbert, and M. Turan, "Endoslam dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos," *Medical Image Analysis*, vol. 71, p. 102058, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841521001043