SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I12P133

Volume 11 Issue 12, 361-374, December 2024
© 2024 Seventh Sense Research Group®

Original Article

Optimizing Retinal Surgery: Reinforcement Learning for Enhanced Microscope-Assisted Robotics

Reena S. Rajan[1], H. Vennila[2]

[1]*Department of Electronics and Communication Engineering, Noorul Islam Centre for Higher Education, Tamil Nadu, India*
[2]*Department of Electrical and Electronics Engineering, Noorul Islam Centre for Higher Education, Tamil Nadu, India*

[1]*Corresponding Author : reena.s.rajan91@gmail.com*

Received: 18 October 2024 Revised: 19 November 2024 Accepted: 17 December 2024 Published: 31 December 2024

Abstract - *Recent progress in ophthalmology provides advanced operating rooms with surgical robots and microscopes. Integrating these tools has a significant impact on the field of retinal surgery. Traditional retinal surgeries were often limited by the risk of tremors and challenges in maintaining steady control during complex surgical procedures, leading to a higher risk of complications. This study proposes a Reinforcement Learning (RL) approach to control a robotic arm in retinal microsurgery to enhance precision and reduce the inherent risks of this delicate procedure. The proposed model consists of several key elements, such as a robotic surgery arm, a microscope, and RL agents to control the surgical instrument in real-time according to the visual feedback from the microscope. The RL agent employs a Deep Q-Network (DQN) architecture by interacting with the environment through a sequence of actions and rewards to enhance the movement of the robotic arm. The model utilizes a Convolutional Neural Network (CNN) to extract features from images or frames for accurate state representation. The results demonstrated superior performance with an accuracy of 95%, precision of 97%, recall of 96%, and an F1 score of 96%. The simulation results confirm the high precision control of the robotic arm for minimizing complications in retinal surgeries.*

Keywords - *Retina, Robotic retina surgery, Microscope, Reinforcement learning, Robotic arm.*

1. Introduction

A thin layer of light-sensitive tissue located at the back of the eye, the retina converts incoming light into neural signals for the brain to process. Conditions like diabetic retinopathy, retinal detachment, and macular degeneration cause visual impairments or sometimes blindness [1]. The most sensitive and toughest procedure in ophthalmology is retinal microsurgery, which requires high precision and skill from surgeons. The success of retinal surgeries depends upon the ability of surgeons to regulate the tiny instrument inside the eye without causing damage to the surrounding tissues. Over the past few decades, microsurgical technologies have revolutionized the treatment of retinal disorders with enhanced results and quicker recovery times [2]. Advanced surgical skills are required for the retinal microsurgery. The dimensions for retinal microsurgery are beyond the physiological threshold for many individuals. Physiological tremors can occur during surgical procedures with tool positioning errors. Patient mobility during the procedure under monitored anaesthesia is another risk factor, leading to higher complications. The inability to detect the pressure variations among the surgeons significantly impacted their control of potentially injurious factors [3].

Traditional retinal surgeries are performed by highly skilled surgeons with microscopes and microsurgical instruments. One of the primary challenges in retinal microsurgery is dealing with the extremely small area, which necessitates movements within a few micrometers. Additionally, the retina is more prone to damage because of its highly sensitive and delicate structure. However, slight hand movement or any distraction can lead to serious complications. Thus, the demand for technological innovations paves the way for robot-assisted retinal microsurgery to improve surgical results by minimizing risks [4]. Robot-assisted technologies provide more precise and controlled microscopic movements to reduce the risks of intraoperative complications. The robotic system can hold the tool steady for a long period of time, which is impossible for a human surgeon. Routine tasks can be repeatedly performed within well-defined safety boundaries, standardizing care equality among all patients by reducing surgeon variability [5]. Moreover, during their training phase, surgeons use simulators to develop skills that are not a perfect analog to real surgical procedures. If robotic systems were widely employed, simulators could enable precise design to mimic the performance characteristics of surgical robotic systems. This study proposes a reinforcement learning method for microscope-assisted robotic retina surgery arm control. The main contributions of the study are outlined below:

- To control a robotic arm in microscope-assisted robotic retina surgery.
- To develop a reinforcement learning model that can automatically adjust the robotic arm during surgical procedures.
- To demonstrate the effectiveness of reinforcement learning in responding in real-time to visual feedback.

The remaining portion of the paper is arranged as follows: Section 2 provides a literature review of retinal surgery methods and explores the research gap in the proposed model. Section 3 detailed the proposed reinforcement learning for microscope-assisted robotic retina surgery with arms control. Section 4 details the in-depth analysis of the results of handling reinforcement learning. Section 5 furnishes concluding remarks.

2. Related Works

Ebrahimi et al. [6] proposed a state-estimating Kalman Filtering (KF) to enhance the insertion depth and instrument tip position. Beam theory was employed with sclera force measurement and robot Forward Kinematics (FWK) to localize the tooltip. The method combined the sensor measurements and robot FWK for insertion depth measurement. The outcomes indicated that after applying KF, the insertion depth and instrument tip position had improved by 94% and 77%, respectively. The study was limited by the reliance on KF under non-linear conditions. Zhou et al. [7] developed 3D navigation of a microsurgical instrument using spotlight projection. The feasibility of the model was assessed by employing a Remote Center of Motion (RCM) constraint using the Steady-Hand Eye Robot (SHER). This model achieved an average tracking error of 0.013 mm compared to manual tracking. However, the study presented challenges due to the changes in the lighting conditions, which affected the precision of the tracking algorithm.

Cereda et al. [8] evaluated an instrument integrated Optical Coherence Tomography (iiOCT) based on a distance sensor for robotic vitreoretinal surgery. The intraoperative and preoperative OCT images were compared by processing the images to assess the SNR ratio at different distances and angles. The results demonstrated the model’s accuracy, with a thickness value difference of less than 5% compared to the measured preoperative OCT. The key limitation of the study was the reduced signal accuracy when imaging the area of excessive retinal curvature. Chatterjee et al. [9] explained the advancements in robot-assisted surgeries. Modern robotic surgery systems are designed with highly sensitive robotic arms and miniaturized surgical instruments for precise visualization while reducing hand tremors. The revolution of artificial intelligence enhanced complex surgical procedures. However, the cost and maintenance of the robotic system and the availability of trained surgeons posed major challenges. Birch et al. [10] used trocar localization with a micro-camera mounted on the surgical forceps for robot-assisted vitreoretinal surgery. The model tracked both ArUco markers attached on either side of a trocar, with the trocar position at the estimated midpoint between the markers. The findings revealed an RMSE of 1.82 mm and 1.24 mm for marker localization and trocar localizations, respectively. The study was limited by its potential for misaligning markers during surgery. Jian et al. [11] introduced a parallel robot for minimally invasive eye surgery with RCM. Both forward and inverse kinematic models were analyzed, which showed partially decoupled motion, and dimension optimization was conducted.

A prototype was developed with precision ranging from 7 ± 2 μm to 30 ±8 μm and accuracy from 21 ±10 μm to 568± 374 μm. The complexity of manufacturing and assembly processes limited the study. Wang et al. [12] designed an image-guided automatic control method for posterior segment eye surgery. The study introduced a tip detection network (Net-SR), which calculates the coordinates of the Tips of Surgical Forceps (ToSF) and Tips of Shadow (ToS). The results demonstrated superior performance over traditional key point detection with a 41.7% improvement in optimal speed. Jacobsen et al. [13] compared robot-assisted and manual vitreoretinal surgery using a surgical simulator in virtual reality. The method involved 10 numbers of surgeons, each representing the vitreoretinal and ophthalmic categories. The experiments utilized the Eyesi virtual-reality simulator. The results showed that the robot-assisted surgery was time-consuming with greater precision value and lesser tissue damage. However, the simulation setup was incapable of spontaneous actions during surgery, which has limitations in real-life surgeries. He et al. [14] used an automatic light pipe actuation model for robot-assisted retinal surgery.

The target region was automatically illuminated utilizing a light pipe attached to the robotic arm and a hybrid force-velocity controller. The results demonstrated the model’s effectiveness in illuminating the target area with negligible offset and average scleral forces of less than 50 mN. However, registering the eyeball in the robot frame before surgery is necessary to achieve the desired area. Guo et al. [15] introduced a hybrid brain-computer interface with simultaneous and sequential modes having steady-state visual eye tracking in a virtual reality environment. The study utilized decision fusion of Electroencephalography (EEG) and eye-gaze in simultaneous mode, whereas no calibration was done in sequential mode. The study’s limitations included the potential challenge of ensuring robustness across various diverse real-world environments. Kim et al. [16] proposed an efficient imaging system using intraoperative OCT and microscope, providing real-time depth feedback. A CNN was employed to segment task-relevant information. Patient care was satisfied using trajectory generation by employing model predictive control. The study demonstrated its effectiveness by automating Subretinal Injection (SI) through 30 trials on pig eyes with an accuracy of 26 ± 12µm. These reviewed studies demonstrate some great advancements in robot-assisted eye surgery with integrated imaging systems, navigation methods, and sensor-based controls that improve precision and depth perception. However, several gaps still remain. Besides, comprehensive validation of these systems in real-life, unpredictable surgical environments is also missing. Besides, high manufacturing complexity and the costs of the systems limit the wider adoption of these technologies. Future research should thus focus on enhancing robustness, handling dynamic and nonlinear conditions, and making the entire process smooth for integrating robotic systems into practical clinical applications of autonomous surgical technologies.

3. Materials and Methods

Robot-assisted retinal microsurgery is crucial because slight hand movements of surgeons during surgical procedures can cause serious complications due to the extreme sensitivity of the retina. Traditional procedures are often limited by the control and stability of micro-scale movements, which can be mitigated by the enhanced capability of robotic arms for autonomous decision-making during complex surgical procedures. This study incorporates reinforcement learning with microscope-assisted robotic arm control to achieve precise real-time surgical procedures. Figure 1 illustrates the block visualization of the suggested model.

3.1. System Overview

The system for robot-assisted retinal microsurgery in this study consists of several key elements, such as a robotic surgery arm, a microscope, and reinforcement learning agents. Each component plays a major role in efficient surgery with high precision. The robotic surgery arm is the physical arm that needs to be controlled. Multiple actuators and joints that can move in specific directions are featured to perform various ranges of motions for different surgical procedures.

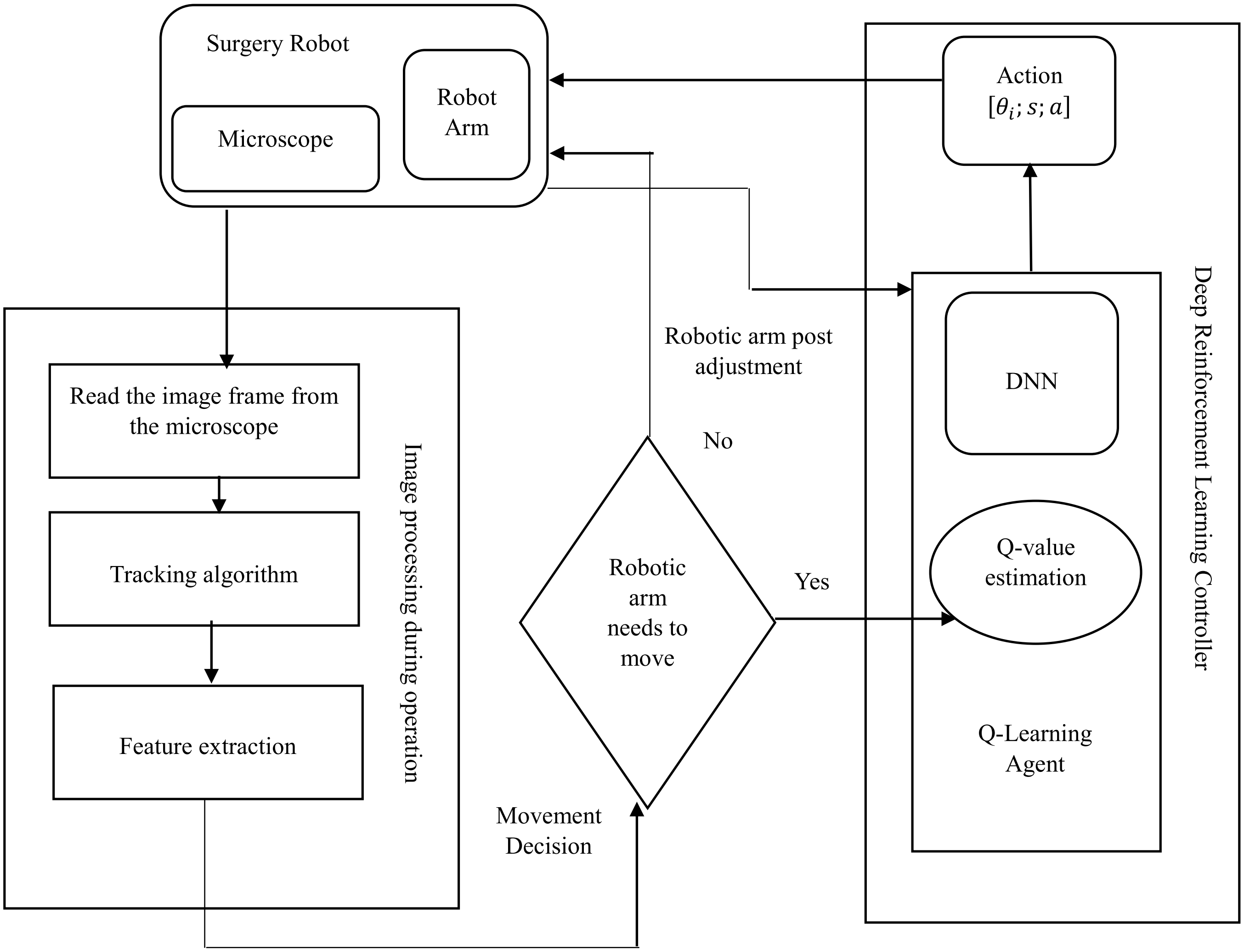

Surgery Robot
Microscope
Robot Arm
Action
$[\theta_i; s; a]$
Read the image frame from the microscope
Tracking algorithm
Feature extraction
Image processing during operation
Robotic arm post adjustment
No
Robotic arm needs to move
Yes
Movement Decision
DNN
Q-value estimation
Q-Learning Agent
Deep Reinforcement Learning Controller

Fig. 1 Block diagram of the suggested model

The movement of a surgical robotic arm is categorized into passive and active parts. The first four joints comprise passive parts for the arm's preoperative positioning. These joints are not actively engaged during the procedures. Using a master manipulator, the doctors control the last four joints, which are active parts, to adjust the position of the microscope or surgical instrument.

Figure 2 illustrates the structure of the surgical robot arm. Another essential component of a robotic surgery system is the microscope. Microscopes capture visual feedback through images or video to guide the robotic arm. This feedback helps detect the current position of the robotic arm by monitoring the surgical area during the surgical procedure. A reinforcement learning agent is employed to learn from the environment and adapt to the feedback from the microscope accordingly to execute movements for the robotic arm.

3.1.1. Surgical Instrument

The tracking and control of the surgical instruments are facilitated by adding simple and specific markers to them. The pose of the surgical instrument is indirectly determined by identifying the pose features of the marker, thereby reducing the increased workload and time required to identify the entire surgical instrument. The marker is designed in accordance with the rotation invariance of the circular feature. This circle becomes an ellipse during rotation without changing the long axis size due to the mapping relationship. Thus, to accurately determine the surgical instrument's orientation and position, the circular marker's long axis represents the size of the surgical tool, and the coordinates of the circular center represent the position information of the tools. The marker setting of the surgical instrument is illustrated in Figure 3.

3.1.2. Visual Field Adjustment of Microscope

Markov Decision Process (MDP), a common RL model to guide decision-making processes, is utilized to adjust microscopic posture automatically [17]. Five tuples represent the MDF $\{S, A, R, P, \gamma\}$, where S is a set of states representing the visual feedback from the microscope, A is the set of actions representing the movements of the robotic arm, R is the reward function guiding the RL agent, P is the transition probability between the new and current states, and γ is the discount factor.

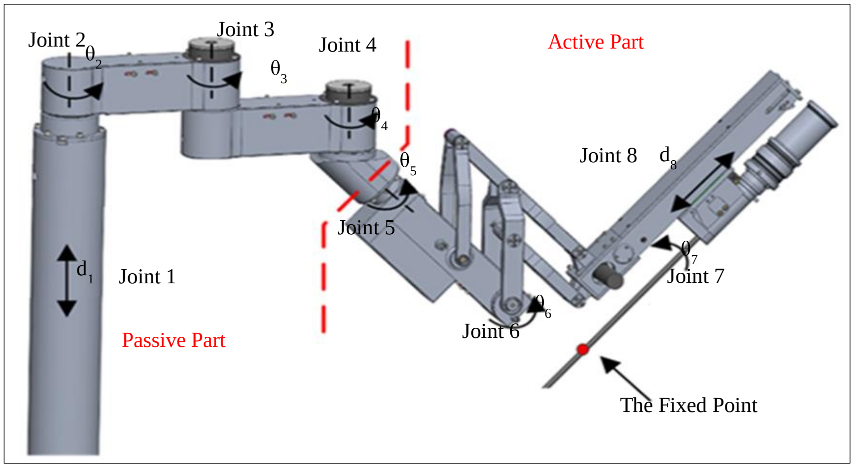


Fig. 2 Structure of the surgical robotic arm

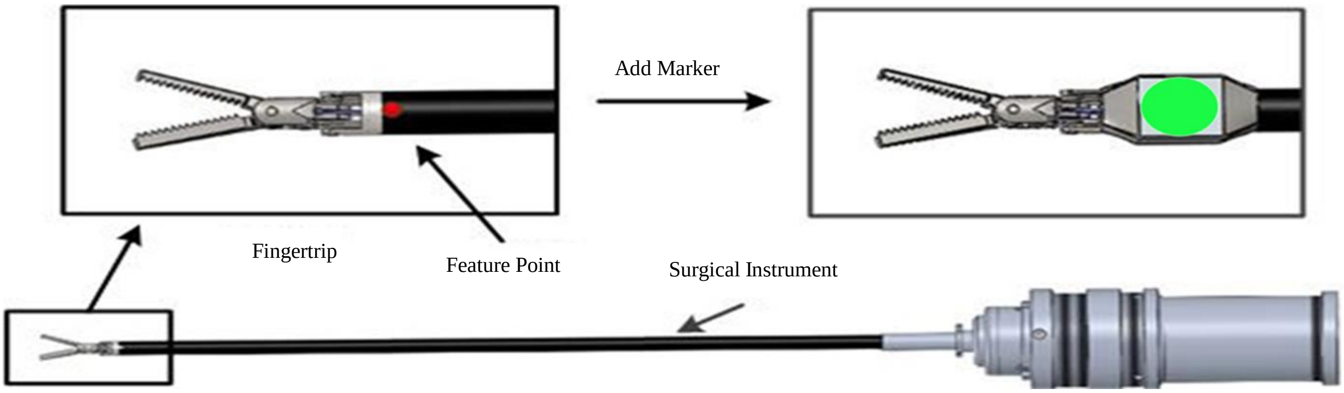


Fig. 3 Marker setting of surgical instruments

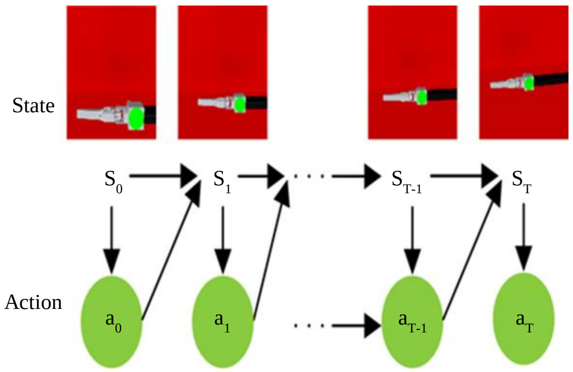


Fig. 4 Microscopic visual field adjustment

Figure 4 provides the diagrammatic representation of the microscopic visual field adjustment. RL agent aims to maximize the total discounted reward (G) as expressed in Equation (1).

$$G(t) = \sum_{j=1}^{\infty} \gamma^j R_{t+j+1} \quad (1)$$

Where t represents the current time step, γ is the discount factor, and R_{t+j+1} is the received reward at $t+j+1$ time step. The Q-function defined by Equation (2) represents the expected future rewards during an action a in a state s under a policy π.

$$Q(s,a) = E_\pi[G_t | S_t = s, A_t = a] \quad (2)$$

Where S_t and A_t are the current state and action, respectively, with expectation E that maps states to action. The maximum action value function defined by Equation (3) is obtained by maximizing the expected future rewards from the next state s' and action a'.

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) max_{a'} Q^*(s',a') \quad (3)$$

Equation (4) derived the optimal policy that maximizes the expected return.

$$\pi^*(s) = arg\ max_{a'} Q^*(s,a) \quad (4)$$

Thus, by MDP, the system makes an action decision according to the visual feedback from the microscope, adjusts the microscopic pose, and updates the policy to maximize future rewards for the automatic visual field adjustment of the microscope.

3.2. Feature Extraction Using Proposed CNN Model

Initially, robotic arm position and microscopic feedback are the primary inputs for state representation. The microscopic feedback consists of images or video frames taken using a sensor integrated with the microscope to extract relevant features such as distance, position, and object orientation. The state of the robotic arm position represents current joint angles or the current position and orientation of the arm. It helps to analyze the smooth movement between the positions for more precise adjustments. A CNN is employed to extract relevant features from the visual data captured by the microscope [18].

The basic CNN architecture is given in Figure 5. The input, $X \in \mathbb{R}^{128\times128\times3}$represents the height and width of the image with three RGB channels. The input consists of pixel values corresponding to each channel's position. There are three convolutional layers with increasing filter sizes (32, 64,128) and kernel sizes of (3, 3) to extract features from the images. The convolution operations between the input image and the filter produced feature maps. Mathematically, the convolution operation is given by Equation (5).

$$y[m,n] = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} x[m+1,n+j]\,.\,w[i,j] \quad (5)$$

Where $y[m,n]$ is the feature map having $x[m+1,n+j]$ as input, k is the filter size, and $w[i,j]$ is the weight value at position (i, j). The ReLU activation function is employed to introduce non-linearity. Max pooling layers reduce the spatial dimensions of the feature maps by selecting the minimum values in each 2×2 window. Mathematically, max pooling is expressed in Equation (6).

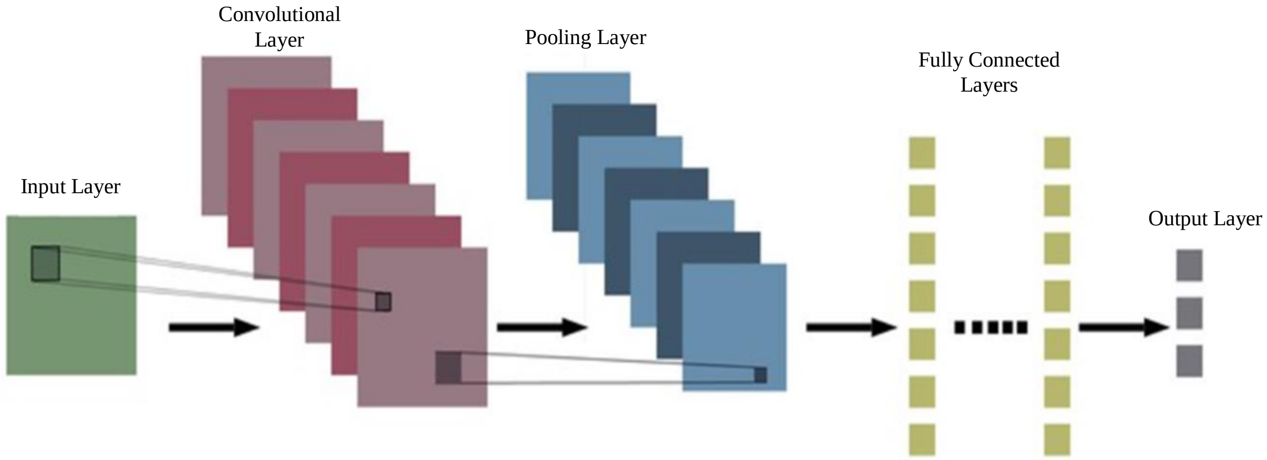


Fig. 5 Basic CNN architecture

$$P(m,n) = max_{i,j}(A[m \cdot s + i, n \cdot s + j]) \qquad (6)$$

The feature map is then flattened into a 1D vector to input into the Fully Connected (FC) layer. Two FC layers with 256 and 128 units were employed, followed by dropout layers to prevent overfitting. Finally, the output layer predicts the Q value for each action, which has units equal to the number of possible actions. Table 1 provides the model summary of the proposed method. Figure 6 provides the proposed model architecture.

Table 1. Proposed model summary

Layer (Type)	Output Shape	Parameters
Input Layer	(None, 128, 128, 3)	0
Conv2D	(None, 128, 128, 32)	896
Maxpooling2D	(None, 64, 64, 32)	0
Conv2D	(None, 64, 64, 64)	18496
Maxpooling2D	(None, 32, 32, 64)	0
Conv2D	(None, 32, 32, 128)	73856
Maxpooling2D	(None, 16, 16, 128)	0
Flatten	(None, 32768)	0
Dense	(None, 256)	8388864
Dropout	(None, 256)	0
Dense	(None, 128)	32896
Dropout	(None, 128)	0
Dense	(None, 4)	516
Total Parameters: 8515524 Trainable parameters: 8515524 Non-trainable parameters: 0		

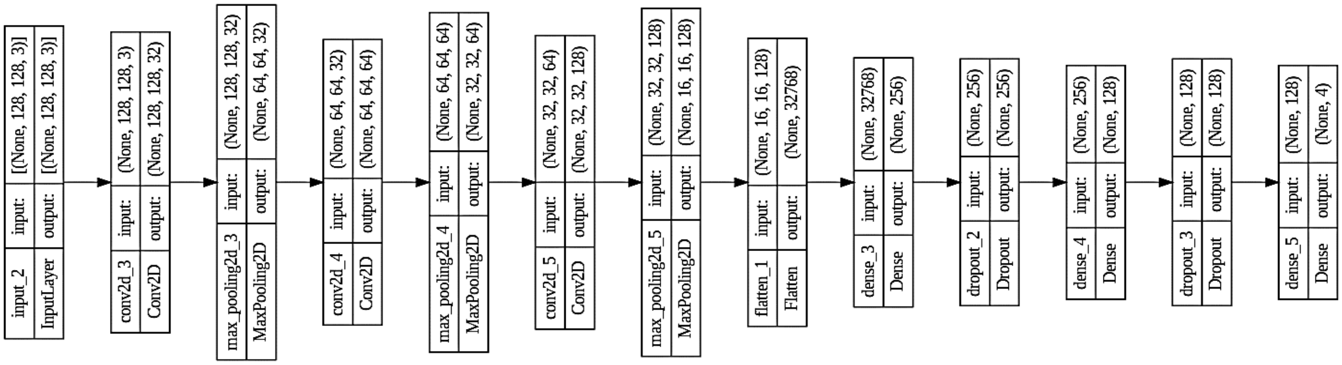

input_2 | InputLayer | input: | output: | [(None, 128, 128, 3)] | [(None, 128, 128, 3)]
conv2d_3 | Conv2D | input: | output: | (None, 128, 128, 3) | (None, 128, 128, 32)
max_pooling2d_3 | MaxPooling2D | input: | output: | (None, 128, 128, 32) | (None, 64, 64, 32)
conv2d_4 | Conv2D | input: | output: | (None, 64, 64, 32) | (None, 64, 64, 64)
max_pooling2d_4 | MaxPooling2D | input: | output: | (None, 64, 64, 64) | (None, 32, 32, 64)
conv2d_5 | Conv2D | input: | output: | (None, 32, 32, 64) | (None, 32, 32, 128)
max_pooling2d_5 | MaxPooling2D | input: | output: | (None, 32, 32, 128) | (None, 16, 16, 128)
flatten_1 | Flatten | input: | output: | (None, 16, 16, 128) | (None, 32768)
dense_3 | Dense | input: | output: | (None, 32768) | (None, 256)
dropout_2 | Dropout | input: | output: | (None, 256) | (None, 256)
dense_4 | Dense | input: | output: | (None, 256) | (None, 128)
dropout_3 | Dropout | input: | output: | (None, 128) | (None, 128)
dense_5 | Dense | input: | output: | (None, 128) | (None, 4)

Fig. 6 Proposed model architecture

3.3. Reinforcement Learning for Robotic Arm Control

The reinforcement learning for robotic arm control involves multiple steps. According to the current state, the robotic arm takes actions and receives rewards based on its performance in fulfilling tasks. RL defines an action space that includes the joint movements for adjusting angles of specific joints, the end effect controller for positioning or orienting the arm's end effector (gripper), and path planning to compute trajectories to reach a specific target or avoid obstacles. Consider the movement of the robotic arm in terms of its joints as in Equation (7).

$$a = [a_1, a_2, \dots \dots a_n] \qquad (7)$$

Where $a_i, i = 1,2,3 \dots n$ defines the movement of i^{th} joint. To understand the current arm configuration, the robotic arm captures the current joint angle or Cartesian coordinates. The rewards are defined as achieving specific tasks, such as task completion, successfully positioning the arm to a desired location or orientation, and reducing error to achieve finer control. The reward function is given by Equation (8).

$$R(s,a) = -\|p_{target} - p_{arm}\| \qquad (8)$$

where p_{target} is the target position and p_{arm} is the current position of the arm. In the RL process, the agent balances exploration and exploitation for the agent's interaction with its environment to maximize the reward function [19]. The exploration strategy refers to the process of trying new actions to learn. It identifies better actions or policies. On the other hand, exploitation strategy employs learned actions to maximize the reward. The action a is selected using an epsilon-greedy policy (ε -greedy) as expressed by Equation (9).

$$Action(a) = \begin{cases} random\ action & with\ probability\ \epsilon \\ argmax_{a'}Q(s,a') & with\ probability\ 1-\epsilon \end{cases} \quad (9)$$

Where ϵ is the exploration rate, which decreases over time as the agent learns.

Q-learning or Deep Q-Networks (DQN) are used as the learning algorithm. The main goal of Q-learning is to estimate the optimal action-value function $Q^*(s,a)$, which denotes that the expected cumulative reward begins from the start s, taking the corresponding action a by following the optimal policy. The update rule of the Q-value is given by Equation (10).

$$Q(s,a) \leftarrow Q(s,a) + \propto (R(s,a) + \gamma max_{a'}Q(s',a') - Q(s,a)) \quad (10)$$

Where $max_{a'}Q(s',a')$ represents the maximum Q-value for the state s' over possible action a'. The Q-values are iteratively improved until they converge to $Q^*(s,a)$ to satisfy the Bellman optimality equation. By replacing the Q-value table with a neural network, DQN enables learning processes in more complex environments where states and actions are high-dimensional. DQN updates the policy based on the rewards for actions taken in different states. In DQN, the Q-value is defined as Equation (11).

$$Q(s,a;\theta) \approx NN(s,a) \quad (11)$$

Where θ represents the weights of the neural network. A Deep Neural Network (DNN) minimizes the difference between the predicted Q-value and the target value obtained from the immediate reward and the discounted future Q-value. The loss function of DNN is given by Equation (12).

$$L(\theta) = \mathbb{E}_{(s,a,r,s')}[(R(s,a) + \gamma max_{a'}Q(s',a';\theta^-) - Q(s,a:\theta))^2] \quad (12)$$

Where θ^- is the parameter of the target network. In DQN, θ is updated using the gradient of the loss function as given by Equation (13).

$$\nabla_\theta L(\theta) = (R(s,a) + \gamma max_a Q(s_{t+1},a;\theta') - Q(s_t,a_t;\theta))\nabla_\theta Q(s_t,a_t;\theta) \quad (13)$$

For training, the RL agent interacts with the environment over multiple episodes. During training, the agent observes the current state as microscope feedback and arm position, selects suitable actions based on learning or exploring as a policy, and executes the selected actions on the robotic arm.

Algorithm 1. Reinforcement learning for robotic arm control

```
Define environment
    • State space: State space S, including joint angles $\theta_i$, ranging from $S_{min}$ to $S_{max}$.
    • Action space: Action A, where the robotic arm can adjust its joint angle $\theta_i$.
Initialization
Initialize Q-values for state-action pairs: $Q(s,a) \leftarrow 0 \quad \forall s \in S, a \in A$
Set the exploration rate $\epsilon$.
Initialize the target network parameter $\theta^-$.
        Set learning rate $\propto$, discount factor $\gamma$
Training loop
Begin
    1. For each episode:
        • Initialize the state s
    2. For each time step within the episode
        • Select an action using $\epsilon$ greedy policy using Equation (9).
        • Execute the action a on the robotic arm and observe the new state $s'$and reward R (s, a).
        • Store the transition (s, a, R (s, a), $s'$) in the replay buffer.
        • Sample a mini-batch of transitions from the replay buffer.
        • Update Q-values using Bellman Equations.
            For Q-learning: Update Q-value using Equation (10).
            For DQN: Compute the target value, $y = R(s,a) + \gamma max_{a'}Q(s',a';\theta^-)$
                Update $\theta$ by minimizing the loss function using Equation (12).
                Compute the gradient of the loss function using Equation (13).
        • Update the target network parameter $\theta^-$
        • Update the state to $s'$for the next time step.
    3. Decaying exploration rate.
End
```

The agent receives rewards as feedback on how well the arm achieved the task objective. The policy (Q-values or neural network weights) was updated to improve the performance over time. Once trained, the RL agent can autonomously control the robotic arm based on real-time feedback from the microscope.

Based on new microscopic images or frames as state information, it continuously adjusts its action to maintain accuracy and precision in surgical tasks. Through continuous learning about environmental interactions and optimizing actions, the model achieves adaptive control over the robotic arm for retina surgery with high responsiveness in dynamic environments. The RL algorithm for robotic arm control is given below.

3.4. Surgery Simulation Setup

The study utilized the crossed platform robotic simulator V-REP for simulation to validate the proposed method [20]. The surgery setup comprises a microscope arm and a surgical instrument, as shown in Figure 7. A visual sensor is mounted to the microscopic arm manipulator to capture RGB images.

The background color of the software is set to red to simulate a real surgical environment. At the beginning of every training session, the microscopic arm is initialized to the same position, and the surgical instrument is aligned in the center of the microscopic view. The robotic arm carries out a series of movements to accurately position the surgical instrument, which moves randomly in three-dimensional space to reach the desired location.

Hyperparameters are crucial configuration parameters that specify how a deep learning framework operates and functions during training. As demonstrated in Table 2, hyperparameters are user-specified prior to training, in contrast to the model’s parameters, which are determined by the data.

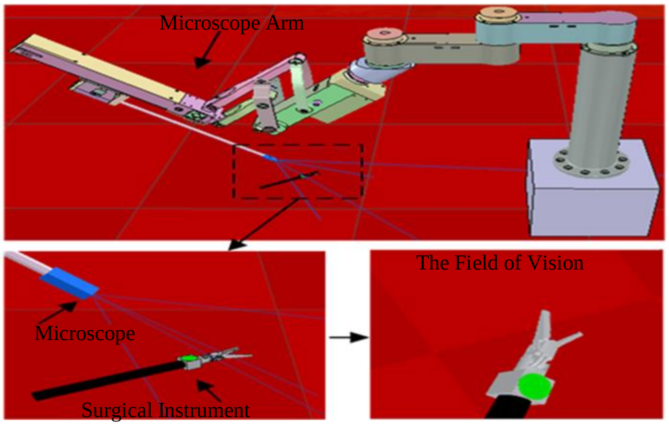


Fig. 7 Surgery setup

Table 2. Hyperparameter specifications

Hyperparameters	Values
Number of epochs	1000
Activation function	ReLU
Batch size	64
Dropout	0.5

4. Results and Discussion

Understanding the effectiveness and learning trends of the suggested model depends on the accuracy and loss plots. The accuracy plot graphically illustrates the model’s capacity to consistently predict data labels throughout training iterations on both the training and validation datasets. Figures 8 and 9 illustrate the proposed model’s accuracy and loss plot. The increase in the accuracy curve during the training and validation phases indicates the effectiveness of RL agents in improving the controlling capabilities of robotic arms, whereas the decreasing loss denotes the accurate predictions of RL agents. The correlation between the model’s predictions and the actual labels is evaluated to assess the model’s effectiveness during training. A loss plot depicts the progression of the model’s loss function over various iterations or epochs throughout the training process. This declining trend indicates that the model’s capacity to reduce prediction mistakes is enhanced over time. Reduced loss values signify superior concordance between the model’s predictions and the actual labels in the training dataset. Analogous to accuracy, variations in loss values are evident throughout epochs, indicating the model’s reaction to changes in the training data and optimization procedure.

A highly effective method for evaluating the accuracy of the proposed model is using a confusion matrix. The matrix provides a systematic overview of the model’s performance by comparing its predictions with the actual labels across multiple classes. It organizes the findings in a tabular format, with rows denoting the real labels and columns indicating the predicted labels. Each cell in the matrix signifies the frequency of instances in which the model’s predictions align with or diverge from the actual labels.

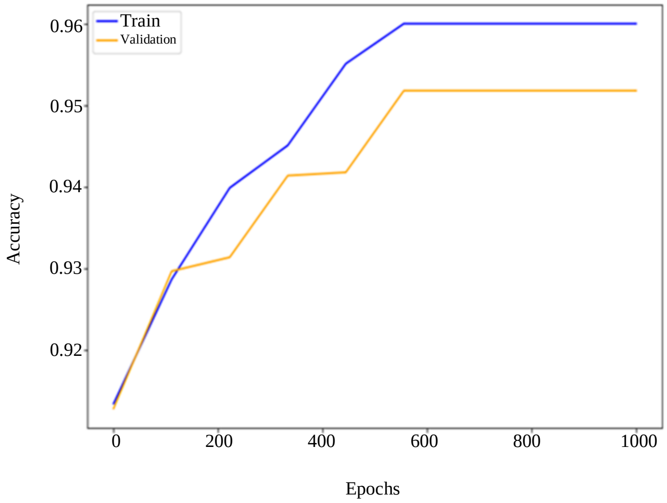


Fig. 8 Accuracy plot of the proposed model

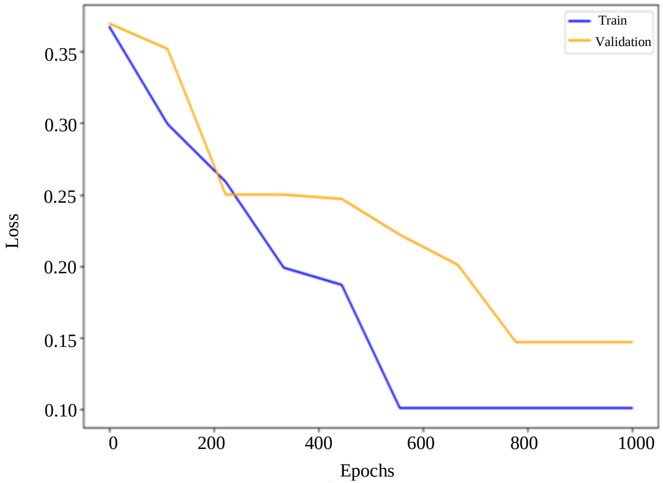


Fig. 9 Loss plot of the proposed model

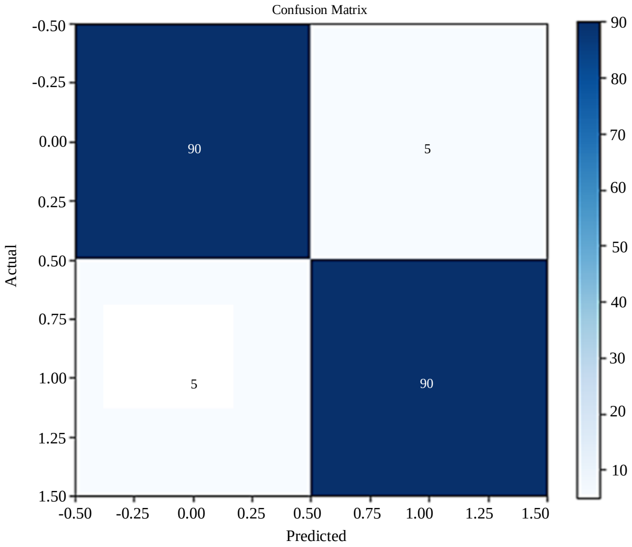


Fig. 10 Confusion matrix

The confusion matrix is divided into four quadrants, with the diagonal elements indicating accurate predictions and the off-diagonal elements signifying misclassifications. Figure 10 represents the confusion matrix of the proposed model with predicted and actual outcomes. This visual representation identifies the agent's ability to classify actions correctly. Performance indicators obtained from the confusion matrix provide a comprehensive assessment of the proposed model's effectiveness. To comprehensively assess the efficacy and operational efficiency of the proposed model, the four principal metrics employed are F1-score, accuracy, precision, and recall. These metrics, based in the principles of False Positive (FP), False Negative (FN), True Negative (TN), and True Positive (TP), are crucial for evaluating the model's efficacy. The mathematical formulations for these performance parameters are shown in Equations (14), (15), (16), and (17).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

$$Precision = \frac{TP}{TP+FP} \tag{15}$$

$$Recall = \frac{TP}{TP+FN} \tag{16}$$

$$F1 - score = 2 \times \frac{precision \times Recall}{Precision+Recall} \tag{17}$$

The obtained performance metrics, as shown in Figure 11, highlight the exceptional efficacy of the developed model. The proposed RL for microscope-assisted robotic retina surgery arm control achieved an accuracy of 95%, indicating the model's efficiency in effectively controlling the robotic arm in the surgical area. 97% precision denotes accurate predictions by minimizing errors affecting impairments to the retina's highly sensitive tissue. With a recall rate of 96%, the model provides accurate actions with respect to the state for increasing surgical safety. 96.5% F1score reflects a balance between precision and recall, indicating the effectiveness of RL. These metrics collectively determine the efficiency of the proposed model for retina surgery with high adaptability and fine control of the RL agent. Figure 12 shows the rewards over episodes. A rising trend indicates the effective learning of agents to achieve task goals. Figure 13 plots the number of steps (length) of each episode. The decreasing trend in the plot suggests the agent's learning efficiency over time.The changes in the states and actions taken by the agent over time are visualized in Figure 14, and it helps to understand how the agent explores the state space and adapts its actions to achieve optimal control. The smooth transitions in the plot indicate efficient learning and effective actions. The line plot of actions the agent takes over time steps within an episode indicates that the agent has learned a stable policy for reaching and maintaining the task goal. During early training, variations in actions show exploration, while consistent actions indicate exploitation of learned policies.

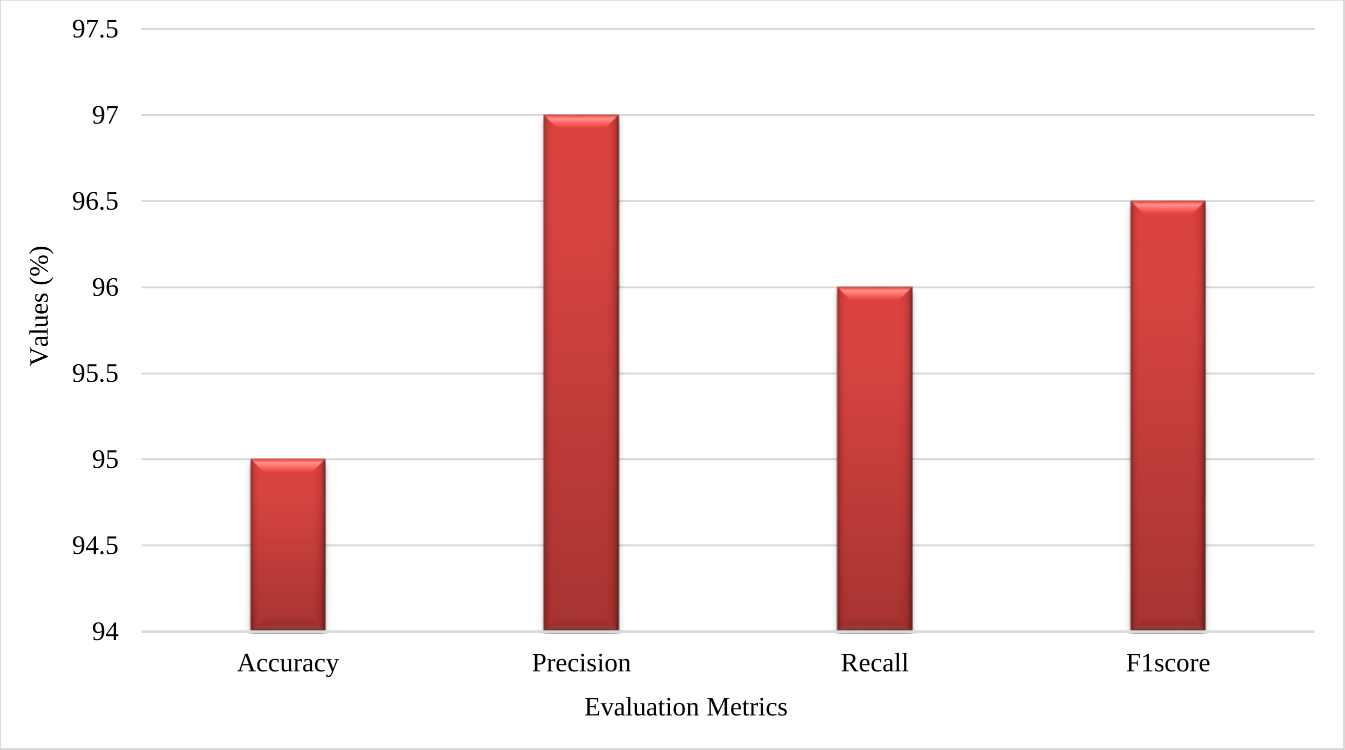


Fig. 11 Performance metrics

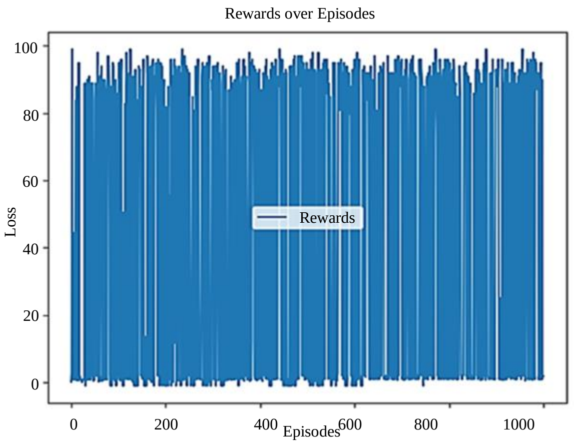


Fig. 12 Rewards over episodes

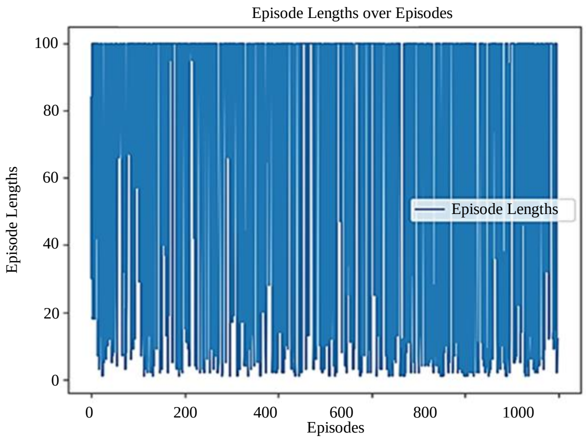
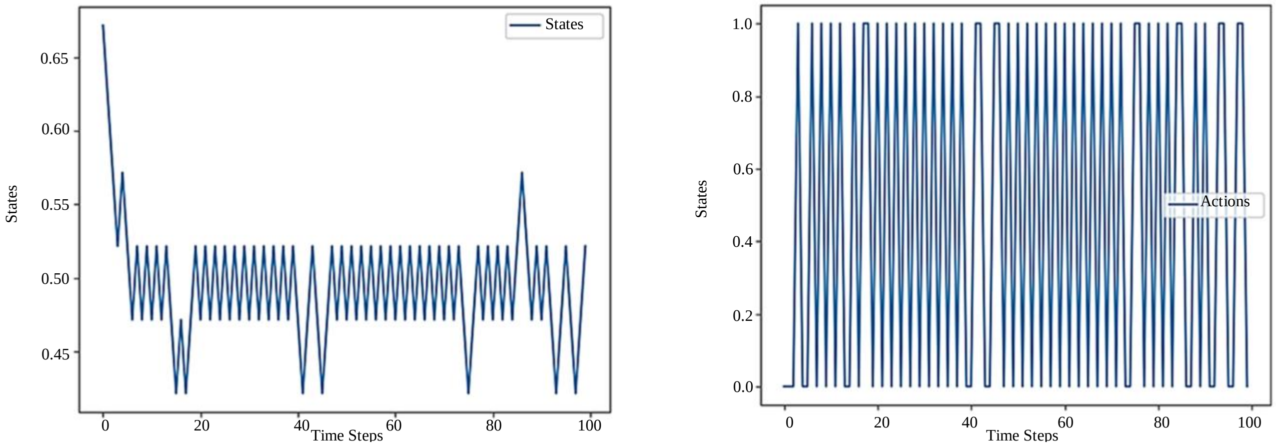


Fig. 13 Episode lengths over episodes



(a) (b)

Fig. 14 (a) Time step vs. States, and (b) Time step vs. Actions.

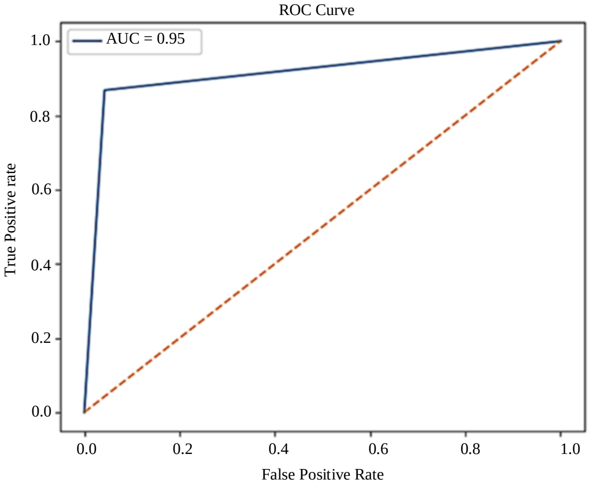


Fig. 15 ROC curve

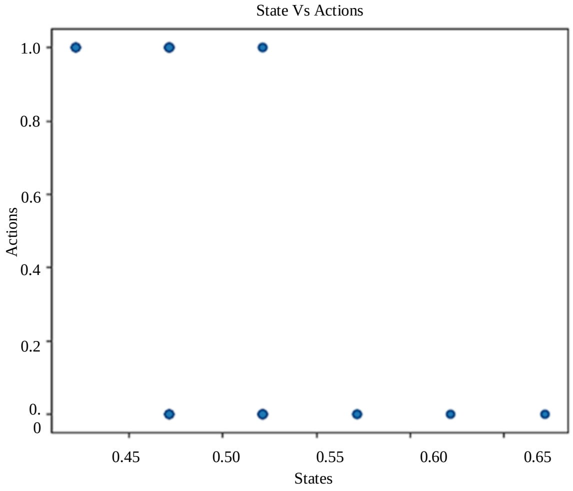


Fig. 16 States vs. Actions Scatter Plot

The Receiver Operating Characteristic (ROC) curve is illustrated in Figure 15, which plots the TP rate against the FP rate. The Area Under the Curve (AUC) summarizes the performance of the ROC curve. A high value of AUC indicates superior performance in distinguishing between the actions. The proposed model has a 0.95 AUC. The scatter plot of states and the corresponding actions taken by the agent is given in Figure 16. This plot indicates that certain states consistently lead to specific actions, showing learned behavior patterns. Due to exploration, early training shows a wider spread, whereas later training shows tighter clustering due to learned policies. Figure 17 illustrates the histogram of states and actions, showing the distribution of states and actions taken by the agent. It reveals the probability of agent visits to different states, with more visits to the goal state indicating effective learning. The frequency of actions with more frequent optimal actions indicates learned policies. The heatmap of Q-values for state-action pairs is illustrated in Figure 18. It visually represents a learned policy, revealing preferred actions for each state. A high Q-value indicates optimal actions for obtaining higher rewards in specific states. The central peak of the state distribution illustrated by Figure 19(a) shows a high frequency at the state of around 0.5, indicating successful learning with frequent agent visits. A wide spread of states during early training suggests extensive exploration. As the training progresses, the spread narrows, indicating more focus on the goal state. The concentration of state visits around that state over time illustrates the agent's efficiency in reaching and maintaining the goal position. As depicted by Figure 19(b), the action distribution exhibits a skewed distribution with a high frequency for action 1 since the agent prefers this action, possibly because it frequently results in higher rewards. A more balanced distribution between actions 0 and 1 might suggest that both actions are equally valuable or that the agent is still exploring.

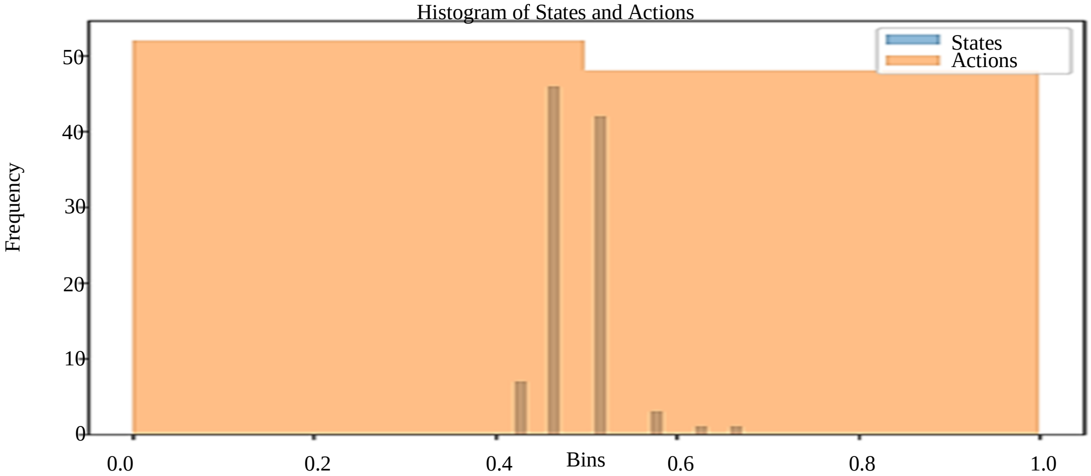


Fig. 17 Histogram of states and actions

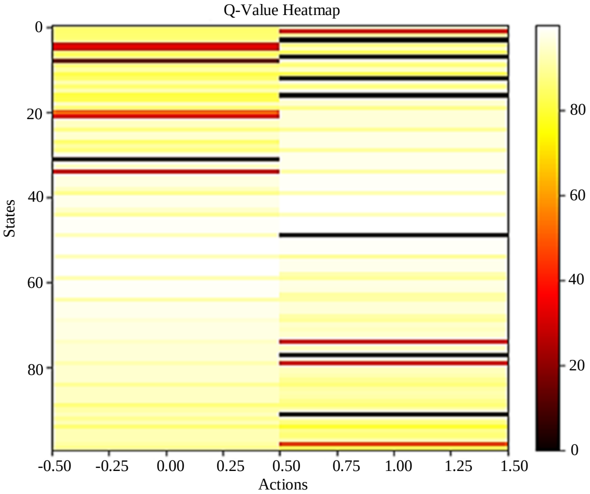
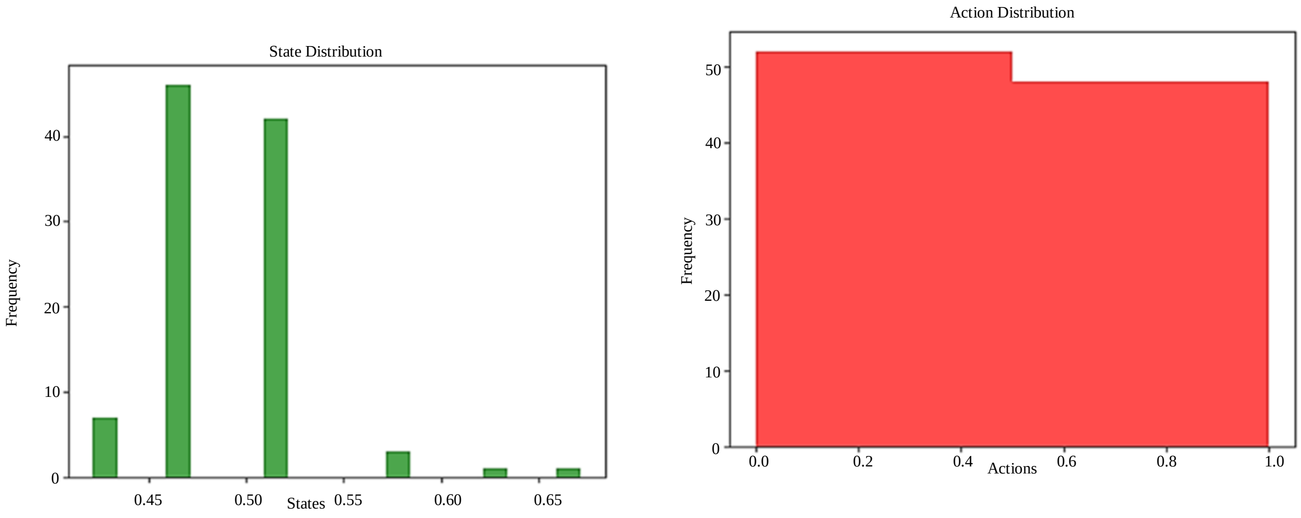


Fig. 18 Q-Value heat map



(a) (b)

Fig. 19 (a) State distribution, and (b) Action distribution.

The action distribution is more uniform during the initial episodes, indicating that the agent explores different actions to understand their effects. The distribution might skew towards those actions as the agent learns which actions lead to better rewards. Once the agent has converged on an optimal policy, the learning behavior of the agent is reflected, showing a clear preference for the optimal actions in the action distribution.

5. Conclusion

This study proposes a novel RL-based method for controlling a microscope-assisted robotic arm in retinal microsurgery, aiming to increase precision and safety in this complex procedure. The proposed system can achieve autonomous adaptation and optimization in the motion of the robotic arm using a DQN with a CNN to process real-time visual feedback. The evaluation of the model underlines its effectiveness with high performance metrics: 95% accuracy, 97% precision, and 96% recall, showing its capability for controlling the surgical instrument with high precision and reducing risks. The RL agent can learn a strategy by which higher rewards are received, and loss decreases over time, showing its efficiency in balancing exploration and exploitation to achieve optimal control. Simulation results confirm the system’s potential for enhancing surgical outcomes through adaptive, precise robotic arm control, thus significantly contributing to the retinal microsurgery research area. This approach promises to enhance surgical accuracy, thus improving patient safety.

Acknowledgments

The author profoundly appreciates the supervisor’s guidance and unwavering support throughout this study.

References

[1] J. Pitcher et al., “Robotic Eye Surgery: Past, Present, and Future,” *Journal of Computer Science & Systems Biology*, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[2] Raffaele Nuzzi, and Luca Brusasco, “State of the Art of Robotic Surgery Related to Vision: Brain And Eye Applications of Newly Available Devices,” *Eye and Brain*, vol. 10, pp. 13-24, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] C.N. Riviere, and P.S. Jensen, “A Study of Instrument Motion in Retinal Microsurgery,” *In Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No. 00CH37143)*, Chicago, IL, USA, vol. 1, pp. 59-60. 2000. [CrossRef] [Google Scholar] [Publisher Link]

[4] Raphael Sznitman et al., “Data-Driven Visual Tracking in Retinal Microsurgery,” *15th International Conference Medical Image Computing and Computer-Assisted Intervention*, France, pp. 568-575, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[5] Mingchuan Zhou et al., “Towards Robotic Eye Surgery: Marker-Free, Online Hand-Eye Calibration using Optical Coherence Tomography Images,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3944-3951, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Ali Ebrahimi et al., “Stochastic Force-Based Insertion Depth and Tip Position Estimations of Flexible FBG-Equipped Instruments in Robotic Retinal Surgery,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1512-1523, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mingchuan Zhou et al., “Spotlight-Based 3D Instrument Guidance for Autonomous Task in Robot-Assisted Retinal Surgery,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7750-7757, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Matteo Giuseppe Cereda et al., “Clinical Evaluation of an Instrument-Integrated Oct-Based Distance Sensor for Robotic Vitreoretinal Surgery,” *Ophthalmology Science*, vol. 1, no. 4, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Swastika Chatterjee et al., “Advancements in Robotic Surgery: Innovations, Challenges and Future Prospects,” *Journal of Robotic Surgery*, vol. 18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jeremy Birch et al., “Trocar Localisation for Robot-Assisted Vitreoretinal Surgery,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 19, no. 2, pp. 191-198, 2024. [CrossRef] [Google Scholar] [Publisher Link] [

[11] Yinglun Jian et al., “A Parallel Robot with Remote Centre-of-Motion for Eye Surgery: Design, Kinematics, Prototype, and Experiments,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 20, no. 4, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ning Wang et al., “Vision-Based Automatic Control of a Surgical Robot for Posterior Segment Ophthalmic Surgery,” *IEEE Transactions on Automation Science and Engineering*, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Forslund Jacobsen Mads et al., “Robot-Assisted Vitreoretinal Surgery Improves Surgical Accuracy Compared with Manual Surgery: A Randomized Trial in a Simulated Setting,” *Retina*, vol. 40, no. 11, pp. 2091-2098, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Changyan He et al., “Automatic Light Pipe Actuating System for Bimanual Robot-Assisted Retinal Surgery,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 6, pp. 2846-2857, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Rongxiao Guo et al., “A Robotic Arm Control System with Simultaneous and Sequential Modes Combining Eye-Tracking with Steady-State Visual Evoked Potential in Virtual Reality Environment,” *Frontiers in Neurorobotics*, vol. 17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Ji Woong Kim et al., “Towards Deep Learning Guided Autonomous Eye Surgery Using Microscope and iOCT Images”, *arxiv*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Csaba Szepesvári, *Algorithms for Reinforcement Learning*, Springer Nature, 2022. [Google Scholar] [Publisher Link]

[18] Manjunath Jogin et al., “Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning,” *In 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, pp. 2319-2323, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Junhyuk Oh et al., “Discovering Reinforcement Learning Algorithms,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1060-1070, 2020. [Google Scholar] [Publisher Link]

[20] Eric Rohmer, Surya P.N. Singh, and Marc Freese, “V-REP: A Versatile and Scalable Robot Simulation Framework,” *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, pp. 1321-1326, 2013. [CrossRef] [Google Scholar] [Publisher Link]