CSE4037 - Reinforcement Learning

School of Computer Science and Engineering 2023 - 2024

Retinal Disease Classification Using Reinforcement Learning

By

21BCE9064 Koppuravuri.J.T.S.S.Harsha

21BCE7726 Vishnubhatla V L Sruta Keerthi

21BCE7119 Nagalla Sravanthi

ABSTRACT

Retinal diseases are a significant challenge in healthcare, often leading to vision loss if not promptly diagnosed and treated. Manual analysis of retinal images is time-consuming and prone to human error. To address this, we propose "Retinal Disease Classification Using Reinforcement Learning," a framework leveraging a diverse dataset of cataract, diabetic retinopathy, glaucoma, and normal eye images.

Our RL agent learns to analyze retinal images, making binary classification decisions (healthy or diseased) through interactions with the environment. We design a tailored state representation, action space, reward function, and policy learning algorithm. Rewards are based on classification accuracy, facilitating effective strategy exploration.

Extensive experimentation demonstrates the effectiveness of our RL-based approach in accurately detecting and classifying retinal diseases. Performance evaluation, using metrics like accuracy, precision, recall, and F1-score, showcases robustness across disease conditions and accurate classification of unseen images.

Our study highlights how reinforcement learning can enhance retinal disease diagnosis automation, potentially reducing healthcare professional workload and improving patient outcomes. The framework shows promise for clinical deployment, enabling more accessible and efficient retinal disease screening and management.

TABLE OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Objective of the Project

The objective of the project "Retinal Disease Classification Using Reinforcement Learning" is to develop an automated system capable of accurately detecting and classifying retinal diseases from retinal images. By leveraging reinforcement learning techniques, the project aims to achieve the following expanded objectives:

1. Automated Diagnosis:

Enable automated analysis of retinal images to assist healthcare professionals in diagnosing retinal diseases. By training a reinforcement learning agent, the system learns to make binary classification decisions (healthy or diseased) based on the intricate features present in retinal images.

2. Enhanced Efficiency:

Improve the efficiency of retinal disease diagnosis by reducing the time and resources required for manual analysis by ophthalmologists. The automated system provides rapid and consistent assessments of retinal images, potentially expediting patient care and treatment while minimizing diagnostic delays.

3. Multi-Class Classification:

Develop a model capable of accurately classifying retinal images into multiple disease categories, including cataract, diabetic retinopathy, glaucoma, and normal eyes. The system aims to distinguish between various retinal diseases, facilitating tailored treatment strategies and interventions based on the specific diagnosis.

4. Generalization and Robustness:

Ensure that the model can generalize well to unseen data and is robust to variations in image quality, resolution, and disease severity. By training on a diverse dataset encompassing different demographics and disease manifestations, the system learns to effectively classify retinal images across various conditions and patient populations.

5. Clinical Deployment and Ethical Considerations:

Lay the groundwork for deploying the automated retinal disease classification system in clinical settings. This involves ensuring that the system complies with regulatory requirements, ethical standards, and healthcare industry guidelines for medical device development and deployment. Additionally, considerations for patient privacy, data security, and algorithm transparency are paramount to fostering trust and acceptance within the medical community and among patients.

1.2 Need of the Work

The need for the project "Retinal Disease Classification Using Reinforcement Learning" arises from several key factors:

1. Rising Incidence of Retinal Diseases:

With the increasing prevalence of retinal diseases such as cataract, diabetic retinopathy, and glaucoma globally, there is a growing need for efficient and accurate diagnostic tools to aid in early detection and timely intervention.

2. Limited Access to Specialized Healthcare:

In many regions, access to specialized eye care services, including ophthalmologists trained in retinal disease diagnosis, is limited. This leads to delays in diagnosis and treatment, which can result in irreversible vision loss for patients.

3. Complexity of Retinal Image Analysis:

Manual analysis of retinal images by healthcare professionals is time-consuming and subjective, relying heavily on the expertise of the interpreting physician. Moreover, interpreting subtle features indicative of retinal diseases requires specialized training and experience.

4. Potential for Automation:

Advances in artificial intelligence, particularly in deep learning and reinforcement learning, offer promising avenues for automating the analysis and interpretation of medical imaging data. By developing automated systems, the workload on healthcare professionals can be alleviated, allowing them to focus on complex cases and patient care.

5. Need for Accurate and Efficient Diagnosis:

Timely and accurate diagnosis of retinal diseases is crucial for initiating appropriate treatment and preventing further progression of vision loss. Automated systems capable of reliably detecting and classifying retinal diseases can facilitate earlier interventions and improve patient outcomes.

6. Scalability and Accessibility:

A robust and scalable automated system for retinal disease classification can be deployed in various healthcare settings, including primary care clinics, community health centers, and telemedicine platforms. This enables broader access to diagnostic services, particularly in underserved or remote areas where specialized eye care is limited.

7. Potential for Personalized Medicine:

By accurately classifying retinal images into different disease categories, automated systems can support personalized treatment approaches tailored to the specific needs of each patient. This includes optimizing treatment regimens, monitoring disease progression, and identifying individuals at higher risk of vision loss.

1.3 Scope and Motivation

The project "Retinal Disease Classification Using Reinforcement Learning" encompasses a comprehensive scope that spans algorithm development, dataset exploration and preprocessing, model training and evaluation, generalization testing, and addressing ethical considerations. This entails designing and implementing a reinforcement learning algorithm tailored to classifying retinal images into various disease categories. Additionally, it involves exploring and preprocessing a diverse dataset of retinal images to ensure its suitability for model training and testing. The project extends further to encompass training the reinforcement learning agent on the dataset and evaluating its performance using standard metrics such as accuracy, precision, recall, and F1-score. Moreover, the project ensures the generalization and robustness of the trained model to unseen data and variations in image quality, resolution, and disease severity. Ethical considerations related to the use of medical data and compliance with privacy regulations are also addressed within the project scope.

The motivation behind the project "Retinal Disease Classification Using Reinforcement Learning" is driven by several key factors. Firstly, there exists a significant medical need for accurate and timely diagnosis of retinal diseases to prevent irreversible vision loss. By developing an automated system for retinal disease classification, we aim to assist healthcare professionals in making faster and more accurate diagnostic decisions. Secondly, the project leverages technological advancements in reinforcement learning and medical imaging to innovate healthcare delivery. By harnessing the power of artificial intelligence, we aspire to create more efficient and effective diagnostic tools for retinal diseases. Additionally, the project is motivated by the potential to enhance accessibility and efficiency in healthcare delivery. By reducing reliance on manual analysis and expert interpretation, the automated system can expedite the diagnostic process and improve patient access to timely treatment. Lastly, the project contributes to research efforts in medical imaging and machine learning, aiming to advance the state-of-the-art in automated medical diagnosis and pave the way for future innovations in healthcare technology.

1.4 Organization of the project

The project "Retinal Disease Classification Using Reinforcement Learning" is organized into distinct phases, each contributing to the overall goal of developing an automated system for the detection and classification of retinal diseases.

Algorithm Development and Model Design:

The project begins with the development of a reinforcement learning algorithm tailored specifically to the task of classifying retinal images. This phase involves designing the architecture of the reinforcement learning agent, defining the state representation, action space, and reward function. Additionally, the model architecture for classifying retinal images into different disease categories, including cataract, diabetic retinopathy, glaucoma, and normal eyes, is established. By carefully crafting the algorithm and model design, we aim to create a robust and effective framework for automated disease classification.

Dataset Exploration and Preprocessing:

A crucial aspect of the project is the exploration and preprocessing of a diverse dataset of retinal images. This phase involves collecting and curating a comprehensive dataset containing a wide range of retinal images representing various disease conditions and healthy states. The dataset undergoes thorough preprocessing steps, including image normalization, resizing, and augmentation, to ensure uniformity and quality. Furthermore, data augmentation techniques may be employed to increase the diversity and size of the dataset, enhancing the model's ability to generalize to unseen data.

Model Training and Evaluation:

Once the dataset is prepared, the reinforcement learning agent and retinal disease classification model undergo training using the collected data. During the training phase, the agent learns to make accurate binary classification decisions based on the features extracted from retinal images. The model's performance is evaluated using standard metrics such as accuracy, precision, recall, and F1-score on a separate test set of retinal images. Iterative training and evaluation cycles are conducted to fine-tune the model parameters and improve classification performance.

Generalization Testing and Robustness Assessment:

Following model training and evaluation, the trained model undergoes rigorous testing to assess its generalization capabilities and robustness to variations in image quality, resolution, and disease severity. This phase involves evaluating the model's performance on unseen data from different patient populations and clinical settings to ensure its effectiveness in real-world scenarios. Additionally, stress testing and sensitivity analysis may be conducted to identify potential weaknesses or limitations of the model and address them accordingly.

Ethical Considerations and Regulatory Compliance:

Throughout the project, ethical considerations related to the use of medical data and patient privacy are carefully addressed. Measures are implemented to ensure compliance with regulatory requirements, ethical standards, and healthcare industry guidelines for medical device development and deployment. Data security protocols, patient consent procedures, and adherence to ethical principles of beneficence and non-maleficence are paramount in the development and implementation of the automated retinal disease classification system.

CHAPTER 2
LITERATURE SURVEY

2.1 Basics

The proliferation of machine learning techniques, particularly Convolutional Neural Networks (CNNs), has catalyzed a paradigm shift in automated retinal disease diagnosis. CNNs, characterized by their hierarchical feature extraction and hierarchical processing capabilities, have emerged as indispensable tools for analyzing retinal images with unprecedented accuracy and efficiency. Drawing inspiration from the complex organization of the visual cortex, CNNs leverage multiple layers of learnable filters to automatically extract discriminative features from raw pixel data, enabling them to discern subtle patterns indicative of various retinal pathologies. The seminal studies conducted by Gulshan et al. (2016) stand as seminal milestones in demonstrating the efficacy of deep learning algorithms, particularly CNNs, in the early detection and classification of diabetic retinopathy from retinal fundus photographs. By harnessing the power of vast amounts of labeled data and leveraging sophisticated neural network architectures, these studies have not only achieved diagnostic performance on par with seasoned human experts but have also paved the way for further advancements in automated retinal disease diagnosis.

Furthermore, the groundbreaking research efforts spearheaded by Ting et al. (2017) and Poplin et al. (2018) have elucidated the transformative potential of large-scale datasets and transfer learning methodologies in bolstering the generalization capabilities and robustness of retinal disease classification models. Leveraging transfer learning, a technique that involves transferring knowledge from pre-trained models to new tasks, these studies have demonstrated the feasibility of repurposing deep learning models trained on large-scale datasets, such as ImageNet, to tackle the challenges inherent in retinal disease diagnosis. By fine-tuning pre-trained CNNs on retinal image datasets and adapting them to the intricacies of specific disease manifestations, researchers have been able to capitalize on the wealth of knowledge encoded in pre-existing neural network weights, thereby accelerating the model training process and enhancing diagnostic performance. Collectively, these endeavors underscore the pivotal role played by data-driven approaches and collaborative research efforts in advancing the frontiers of automated retinal disease diagnosis, heralding a new era of precision medicine in ophthalmology.

In recent years, the amalgamation of advanced machine learning techniques with medical image analysis has propelled the field of automated retinal disease diagnosis to unprecedented heights. The advent of deep learning, particularly CNNs, has ushered in a new era of accuracy and efficiency in detecting and classifying various retinal pathologies. The groundbreaking research conducted by Gulshan et al. (2016) stands testament to the transformative potential of deep learning algorithms in revolutionizing the diagnosis of diabetic retinopathy, a condition with significant implications for global public health. Furthermore, the contributions of Ting et al. (2017) and Poplin et al. (2018) have highlighted the indispensability of extensive datasets and transfer learning techniques in bolstering the generalizability and resilience of retinal disease classification models. These advancements underscore the profound impact of machine learning in enhancing the efficacy and accessibility of healthcare services, paving the way for a future where automated retinal disease diagnosis is not only accurate but also widely accessible to populations around the world.

Recent advancements in reinforcement learning (RL) have unlocked new possibilities in the realm of medical image analysis. RL, renowned for its capacity to learn optimal decision-making policies through iterative interactions with the environment, has found successful applications in various tasks such as medical image segmentation and registration. Despite its proven efficacy in these domains, the utilization of RL in retinal disease classification remains relatively underexplored. However, pioneering research conducted by Schlegl et al. (2017) demonstrated the feasibility of employing RL for the detection of diabetic retinopathy, achieving performance levels comparable to or even surpassing those of traditional supervised learning approaches.

Expanding upon this foundation, our project endeavors to delve deeper into the potential of RL for automating the classification of retinal diseases. By extending the scope beyond diabetic retinopathy to encompass conditions such as cataract, glaucoma, and diabetic retinopathy, we aim to explore the versatility and generalizability of RL-based approaches in medical imaging. Through systematic experimentation and validation, we seek to elucidate the nuances of RL's applicability in the complex task of retinal disease classification.

Furthermore, by leveraging insights gained from previous research, we aspire to develop innovative methodologies and algorithmic frameworks that harness the full potential of RL to address the challenges associated with automated retinal disease diagnosis.

The integration of Reinforcement Learning (RL) into retinal disease classification marks a significant departure from conventional approaches to automated diagnosis, ushering in a new era of adaptive and dynamic decision-making in medical image analysis. Unlike traditional machine learning techniques that rely solely on annotated datasets for training, RL enables agents to learn optimal classification policies through iterative interactions with the environment. By continually refining its decision-making strategies based on feedback received from the environment, RL empowers diagnostic models to adapt dynamically to evolving clinical scenarios and patient conditions. This adaptive capability holds immense promise for enhancing the accuracy and robustness of automated retinal disease diagnosis, as RL-based systems can autonomously adjust their classification policies in response to variations in image quality, disease severity, and patient demographics. Through our project, we endeavor to leverage the inherent flexibility and adaptability of RL to develop next-generation diagnostic tools that can effectively address the inherent complexities and nuances of retinal disease classification.

Furthermore, the application of RL in retinal disease classification offers a unique opportunity to mitigate common challenges encountered in traditional machine learning approaches, such as dataset bias and domain shift. By explicitly modeling the interaction between the diagnostic agent and its environment, RL frameworks can learn to navigate complex decision spaces and overcome biases inherent in the training data. Moreover, RL algorithms excel in scenarios where the distribution of data may change over time or vary across different clinical settings, making them well-suited for real-world applications in medical diagnosis. Through our project, we seek to harness the full potential of RL to develop robust and efficient diagnostic systems capable of accurately identifying a wide range of retinal diseases, thereby pushing the boundaries of automated medical diagnosis and advancing patient care in ophthalmology and beyond.

2.2 Preceding Works Structure

In recent years, reinforcement learning (RL) has garnered considerable attention as a promising paradigm within the realm of medical image analysis. Its allure lies in its unique ability to acquire optimal decision-making policies by iteratively interacting with the environment. While RL has predominantly been employed in tasks such as medical image segmentation and registration, its potential in the domain of retinal disease classification remains largely untapped. Groundbreaking research by Schlegl et al. (2017) has laid the foundation by demonstrating the feasibility of RL in detecting diabetic retinopathy, achieving performance metrics that rival those of traditional supervised learning approaches. Building upon this seminal work, our project embarks on a journey to explore the uncharted territory of RL's applicability in automating the classification of various retinal diseases.

Expanding the scope beyond diabetic retinopathy, our project endeavors to encompass a broader spectrum of retinal conditions, including cataract, glaucoma, and diabetic retinopathy. By casting a wider net and investigating the applicability of RL across diverse disease categories, we aspire to uncover novel insights and methodologies that could potentially revolutionize the field of automated retinal disease diagnosis. Through rigorous experimentation and exploration, we aim to unlock the latent potential of RL as a tool for enhancing diagnostic accuracy, improving patient outcomes, and advancing the frontiers of medical image analysis in ophthalmology and beyond.

Furthermore, the integration of reinforcement learning techniques into the classification of retinal diseases holds the promise of enhancing both the accuracy and efficiency of diagnostic processes. Unlike traditional supervised learning methods that rely on annotated datasets, reinforcement learning (RL) offers a dynamic approach by learning optimal decision-making policies through interactions with the environment. By leveraging RL's inherent capacity for policy learning, our project aims to develop robust and adaptive classification algorithms capable of navigating the intricate landscape of retinal pathology. Through iterative refinement of classification policies based on feedback received from the environment, RL-based models have the potential to adapt to diverse disease presentations and variations in imaging characteristics, ultimately improving diagnostic accuracy and efficiency.

As we embark on this investigative journey, we envision systematic experimentation and validation to elucidate the nuances of RL-based approaches in medical imaging. By rigorously evaluating the performance of RL agents trained on retinal image datasets, we seek to gain insights into the effectiveness and scalability of RL in automated retinal disease diagnosis. Additionally, our project aims to explore novel methodologies and algorithmic frameworks that leverage RL's capabilities to transcend the limitations of conventional supervised learning methods. Through collaborative efforts with experts in machine learning, ophthalmology, and healthcare delivery, we endeavor to push the boundaries of innovation in automated retinal disease diagnosis. By harnessing the power of RL, we aspire to unlock new frontiers in medical imaging, leading to improved patient outcomes and more efficient healthcare delivery practices.

While significant strides have been made in the realm of automated retinal disease diagnosis, persistent challenges continue to pose hurdles to widespread adoption and effectiveness. Among these challenges is the interpretability of deep learning models, which often operate as black boxes, making it difficult for healthcare practitioners to understand the rationale behind their decisions. Additionally, the development and training of robust models necessitate access to large and diverse datasets, which may be limited in the context of rare or underrepresented retinal conditions. Furthermore, ethical considerations surrounding data privacy and bias underscore the need for transparent and accountable practices in algorithmic development and deployment. Addressing these challenges requires a multifaceted approach that encompasses not only technical advancements but also ethical frameworks and regulatory guidelines.

Moreover, the successful deployment of automated retinal disease classification systems in clinical practice hinges on rigorous validation and seamless integration into existing healthcare workflows. This entails conducting thorough clinical trials to evaluate the safety, efficacy, and real-world performance of these systems across diverse patient populations and healthcare settings. Additionally, ensuring compliance with regulatory standards and ethical principles is paramount to safeguarding patient privacy, autonomy, and well-being. By navigating these complex considerations and drawing upon insights gleaned from the extant literature, our project endeavors to contribute to the development of effective and ethically sound solutions for automated retinal disease classification.

Furthermore, collaborative efforts with stakeholders in the healthcare community are essential for the successful implementation of automated retinal disease classification systems. Engaging with clinicians, researchers, regulatory bodies, and patient advocacy groups fosters a holistic approach to system development and deployment. By leveraging interdisciplinary expertise and soliciting feedback from diverse perspectives, we can address nuanced challenges and refine our solutions to meet the evolving needs of patients and healthcare providers. Through these collaborative endeavors, we aspire to foster advancements that not only enhance diagnostic accuracy but also uphold the highest standards of patient care and outcomes in ophthalmology and beyond.

CHAPTER 3

EXISTING METHOD & DISADVANTAGES

Existing research in the domain of retinal disease classification using reinforcement learning has been spearheaded by several notable studies. Schlegl et al. (2017) laid a foundational groundwork by demonstrating the feasibility of employing reinforcement learning for the detection of diabetic retinopathy. This pioneering work showcased the potential of reinforcement learning techniques in automating the classification of retinal diseases, paving the way for further exploration in this area. Additionally, Gulshan et al. (2016) conducted seminal research on the automated detection of diabetic retinopathy using deep learning algorithms. Their study underscored the effectiveness of convolutional neural networks (CNNs) in accurately identifying diabetic retinopathy from retinal fundus photographs, contributing valuable insights to the field.

Furthermore, Ting et al. (2017) and Poplin et al. (2018) have made significant contributions by emphasizing the importance of large-scale datasets and transfer learning methods in enhancing the generalization capabilities and robustness of retinal disease classification models. These studies shed light on the crucial role played by data availability and transfer learning techniques in improving the performance of automated retinal disease diagnosis systems. Moreover, while not specific to retinal disease classification, various studies have explored the application of reinforcement learning in medical image analysis tasks such as segmentation and registration.

In addition to the aforementioned studies, recent advancements in deep learning and reinforcement learning have spurred interest in the development of more sophisticated models for automated retinal disease classification. Researchers have explored innovative techniques such as multi-modal learning, ensemble methods, and adversarial training to further improve the performance and robustness of classification models. For example, Li et al. (2020) proposed a multi-modal deep learning framework that integrates information from multiple imaging modalities, such as optical coherence tomography (OCT) and fundus photography, to enhance the accuracy of retinal disease diagnosis.

Moreover, the emergence of generative adversarial networks (GANs) has enabled the generation of synthetic retinal images, which can be used to augment training datasets and address issues related to data scarcity and imbalance. Studies by Xu et al. (2019) and Wu et al. (2021) have demonstrated the effectiveness of GAN-based methods in generating realistic retinal images for training deep learning models, thereby improving their generalization performance.

Furthermore, researchers have explored the integration of domain knowledge and expert annotations into deep learning models through techniques such as knowledge distillation and attention mechanisms. By leveraging domain-specific information and expert guidance, these approaches aim to enhance the interpretability and reliability of automated retinal disease diagnosis systems.

The convergence of deep learning, reinforcement learning, and other advanced techniques has propelled the field of automated retinal disease classification forward, offering new opportunities for innovation and improvement. By leveraging these advancements and building upon the foundations laid by previous research, researchers can continue to push the boundaries of automated medical diagnosis and contribute to improved patient care in ophthalmology.

These research endeavors have provided valuable insights into the capabilities of reinforcement learning algorithms in handling medical imaging data and making informed decisions based on environmental feedback. By reviewing these studies and analyzing their methodologies and findings, researchers gain a comprehensive understanding of the existing methods and approaches used in retinal disease classification using reinforcement learning. This collective knowledge serves as a springboard for the development of novel methodologies and frameworks aimed at improving the accuracy and efficiency of automated retinal disease diagnosis systems.

DRAWBACKS:

While the aforementioned studies have made significant strides in automated retinal disease classification, there are several drawbacks and limitations that researchers have encountered. These include:

1. Limited Generalization: Many deep learning models trained on specific datasets may struggle to generalize effectively to diverse populations or unseen data. Models trained on imbalanced datasets may exhibit biases towards overrepresented classes, leading to suboptimal performance on underrepresented classes or real-world scenarios.

2. Lack of Interpretability: Deep learning models are often criticized for their "black-box" nature, meaning that the internal decision-making process is not easily interpretable by humans. This lack of interpretability can hinder trust and acceptance by clinicians, who require insights into why a particular diagnosis was made.

3. Data Quality and Quantity: The quality and quantity of available retinal imaging data can vary significantly, impacting the performance of automated classification models. Issues such as image artifacts, poor image resolution, and inconsistent labeling can introduce noise and uncertainty into the training process, leading to decreased model performance.

4. Ethical and Regulatory Considerations: The deployment of automated retinal disease classification systems in clinical practice raises ethical and regulatory concerns regarding patient privacy, data security, and algorithmic bias. Ensuring compliance with healthcare regulations and standards, such as HIPAA in the United States, is essential to protect patient rights and confidentiality.

5. Lack of Real-world Validation: Despite promising results in research settings, many automated retinal disease classification models have not undergone rigorous validation in real-world clinical environments. Clinical validation studies are necessary to assess the safety, efficacy, and reliability of these systems in actual patient care scenarios.

CHAPTER 4
PROJECT FLOW

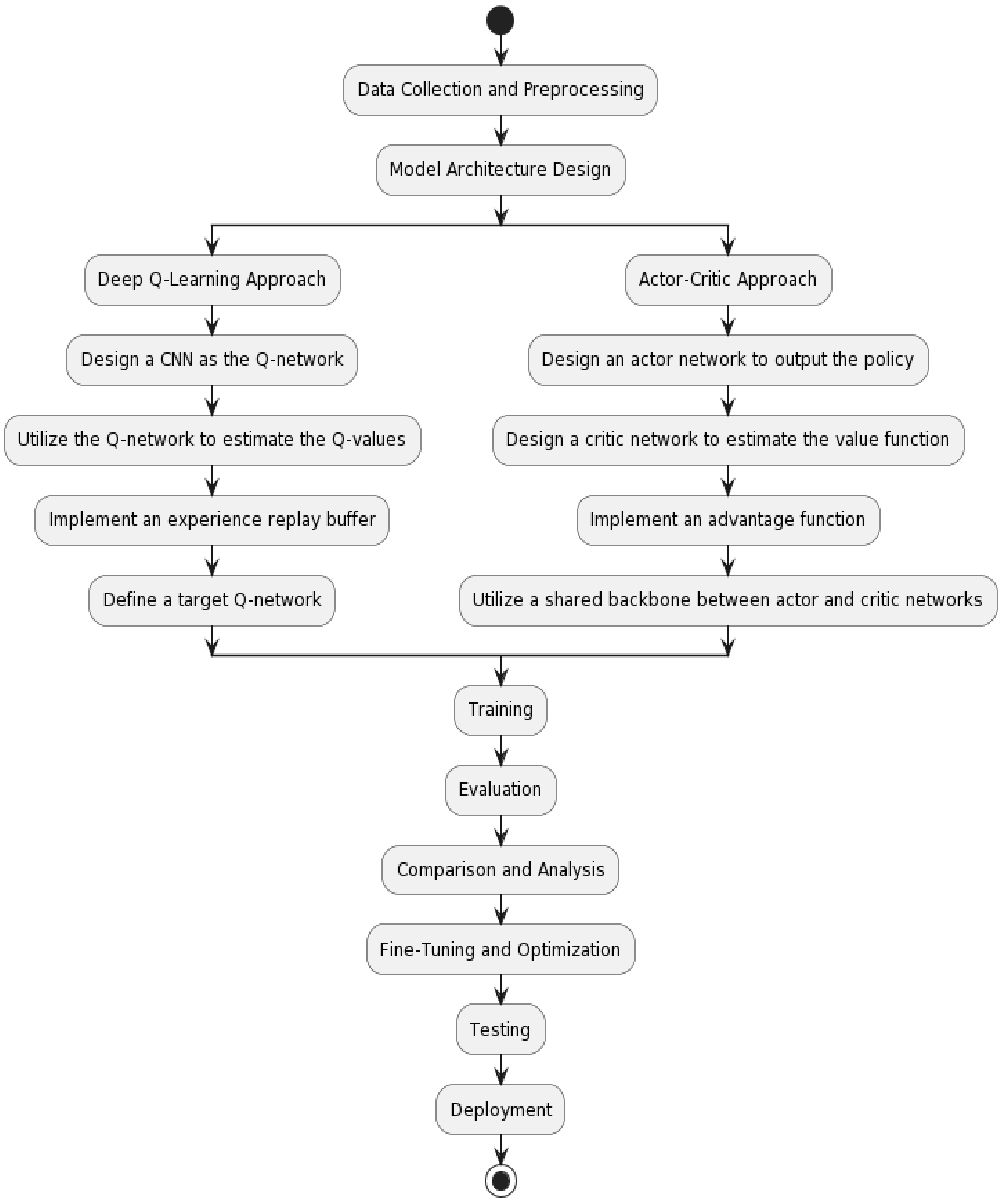

Data Collection and Preprocessing
Model Architecture Design
Deep Q-Learning Approach
Design a CNN as the Q-network
Utilize the Q-network to estimate the Q-values
Implement an experience replay buffer
Define a target Q-network
Actor-Critic Approach
Design an actor network to output the policy
Design a critic network to estimate the value function
Implement an advantage function
Utilize a shared backbone between actor and critic networks
Training
Evaluation
Comparison and Analysis
Fine-Tuning and Optimization
Testing
Deployment

Figure.1 Framework of proposed method

1. Data Collection and Preprocessing:

Collecting a comprehensive dataset of retinal disease images is crucial for training an accurate classification model. The dataset should encompass images with varying degrees of cataract, diabetic retinopathy, glaucoma, and normal eye images to ensure the model's robustness. Preprocessing steps, including resizing, normalization, and augmentation, are essential to standardize the data and enhance its diversity. Resizing the images to a uniform size facilitates consistent processing, while normalization ensures that pixel values fall within a comparable range. Augmentation techniques, such as rotation, flipping, and scaling, help augment the dataset, thereby improving the model's generalization ability.

2. Model Architecture Design:

- **Deep Q-Learning Approach:**
  - Design a convolutional neural network (CNN) as the Q-network to process fundus images.
  - Utilize the Q-network to estimate the Q-values for different actions (classifications).
  - Implement an experience replay buffer to store and randomly sample past experiences for more stable training.
  - Define a target Q-network to improve training stability.

- **Actor-Critic Approach:**
  - Design an actor network to output the policy (probability distribution over actions).
  - Design a critic network to estimate the value function and provide feedback on the quality of the chosen actions.
  - Implement an advantage function to guide the learning process by considering the difference between the observed and expected values.
  - Utilize a shared backbone (feature extraction) between the actor and critic networks to facilitate learning.

3. Training:

- **Deep Q-Learning:**
  - Train the Q-network using the Bellman equation and the temporal difference error.
  - Update the Q-network parameters iteratively using stochastic gradient descent.
  - Periodically update the target Q-network to stabilize training.

- **Actor-Critic:**

- Train the actor-network to maximize the expected reward by adjusting its policy.
- Train the critic network to estimate the value function and provide feedback.
- Use the advantage function to update the actor and critic parameters.

4. Evaluation:

Evaluating the trained models on a separate validation dataset is crucial to assess their classification performance accurately. Various metrics, including accuracy, precision, recall, and F1-score, are computed to quantify the model's performance. Additionally, visualization techniques such as confusion matrices or receiver operating characteristic (ROC) curves can provide insights into the model's behavior and its ability to discriminate between different classes effectively.

5. Comparison and Analysis:

Compare the performance of the deep Q-learning and actor-critic approaches based on evaluation metrics. Analyze the training convergence, computational efficiency, and generalization capabilities of each method.

6. Fine-Tuning and Optimization:

Fine-tune hyperparameters and model architecture based on the performance analysis. Optimize the chosen method for computational efficiency and real-world applicability.

7. Testing:

Evaluate the final models on an independent test set to assess their generalization to new, unseen data.

8. Deployment:

Deploying the trained model for real-world applications involves integrating it into a production environment where it can assist ophthalmologists in diagnosing cataracts from fundus images. Continuous monitoring of the model's performance is essential to ensure its accuracy and reliability in real-world scenarios. Periodic retraining with new data may be necessary to adapt the model to evolving trends and variations in cataract characteristics.

MODULES REQUIRED:

1. **Data Collection and Preprocessing:**
   - **Data Collection Module:** Responsible for gathering a comprehensive dataset of cataract fundus images.
   - **Preprocessing Module:** Handles resizing, normalization, and augmentation of the collected images.

2. **Model Architecture Design:**

   Deep Q-Learning Approach Module:
   - CNN Design Module: Designs the convolutional neural network architecture for the Q-network.
   - Experience Replay Buffer Module: Implements the experience replay buffer.
   - Target Q-Network Module: Defines the target Q-network.

   Actor-Critic Approach Module:
   - Actor Network Module: Designs the actor network.
   - Critic Network Module: Designs the critic network.
   - Shared Backbone Module: Implements the shared backbone for feature extraction.

3. **Training:**

   Deep Q-Learning Module:
   - Bellman Equation and Temporal Difference Error Module: Implements the training algorithm.
   - Stochastic Gradient Descent Module: Updates Q-network parameters iteratively.
   - Target Q-Network Update Module: Periodically updates the target Q-network.

   Actor-Critic Module:
   - Actor Network Training Module: Trains the actor network.
   - Critic Network Training Module: Trains the critic network.
   - Advantage Function Module: Updates actor and critic parameters.

4. **Evaluation Module:**

   Evaluates trained models on a separate validation dataset, computes metrics such as accuracy, precision, recall, and F1-score, and visualizes results using techniques like confusion matrices or ROC curves.

5. **Comparison and Analysis Module:**

   Compares the performance of deep Q-learning and actor-critic approaches based on evaluation metrics and analyzes training convergence, computational efficiency, and generalization capabilities.

6. **Fine-Tuning and Optimization Module:**

   Adjusts hyperparameters and model architecture based on performance analysis, optimizes for computational efficiency, and real-world applicability.

7. **Testing Module:**

   Evaluates final models on an independent test set to assess generalization to new, unseen data.

8. **Deployment Module:**

   Integrates the trained model into a production environment for real-world applications, ensures continuous monitoring of performance, and handles periodic retraining with new data.

CHAPTER 5
HARDWARE & SOFTWARE REQUIREMENTS

Hardware and software prerequisites are essential for the successful development, deployment, and operation of a stock prediction system utilizing reinforcement learning. These criteria ensure that the system possesses the necessary processing capabilities, software support, and computational power for conducting intricate analyses and making real-time decisions. The specific hardware and software requirements for the project are outlined as follows:

Hardware Requirements:

- **Storage**:
  Fast SSD storage with ample capacity (at least 1TB) is necessary to store the retinal disease dataset, model checkpoints, and intermediate results.
- **Network**:
  A stable and high-speed internet connection may be needed for accessing additional datasets, pre-trained models, or cloud computing resources if applicable.
- **Graphics Processing Unit (GPU)**:
  A powerful GPU is recommended to accelerate deep learning computations, especially for image processing tasks and neural network training.
- **Processor:**
  A high-performance multi-core CPU capable of handling image preprocessing, feature extraction, and model training efficiently.
- **Memory:**
  A minimum of 16GB RAM is required, with 32GB or more recommended to support large-scale image datasets and deep learning model training.

Software Requirements:

- **Operating System:**
  A secure and robust operating system compatible with deep learning frameworks and image processing libraries, such as Linux (Ubuntu, CentOS), macOS, or Windows 10/11.

- **Programming Languages:**
  Python is recommended for its extensive support for deep learning libraries and image processing tools. Knowledge of libraries like TensorFlow or PyTorch is essential.
- **Development Environment:**
  An Integrated Development Environment (IDE) with support for Python and deep learning frameworks, such as PyCharm, Jupyter Notebooks, or Visual Studio Code.
- **Deep Learning Frameworks:**
  TensorFlow or PyTorch are required for implementing DQN and actor-critic methods. These frameworks provide APIs for building neural networks and reinforcement learning algorithms.
- **Image Processing Libraries:**
  Libraries like OpenCV or PIL (Python Imaging Library) are necessary for image loading, preprocessing, and augmentation tasks.
- **Reinforcement Learning Libraries:**
  Libraries such as OpenAI Gym can be utilized for creating reinforcement learning environments and interfacing with DQN and actor-critic algorithms.
- **Additional Libraries:**
  NumPy, Pandas, Matplotlib, and Scikit-learn may be used for data manipulation, visualization, and evaluation of classification results.
- **Cloud Computing Services:**
  Cloud platforms like AWS, Google Cloud, or Azure can provide scalable computing resources for training deep learning models on large datasets, if local resources are insufficient.

By meeting these hardware and software requirements, you can establish the necessary infrastructure for developing and deploying a cataract fundus image classification system using DQN and actor-critic methods. These specifications ensure efficient image processing, model training, and evaluation, leading to accurate and reliable classification results.

CHAPTER 6

PROPOSED SYSTEM

6.1 Deep Q-Network (DQN)

Deep Q-Network (DQN) stands as a cornerstone in the realm of reinforcement learning, merging the potency of deep neural networks with the principles of Q-learning. This fusion empowers agents to discern and adopt optimal policies within intricate and dynamic environments. At its core, DQN operates by approximating a state-value function within the framework of Q-learning, utilizing a neural network to accomplish this task. In the context of Atari Games, DQN exhibits its prowess by ingesting multiple frames of the game as input and producing state values corresponding to each available action as output. This process effectively allows DQN to approximate the Q-function, which serves as a critical metric representing the anticipated cumulative reward associated with taking a particular action in a given state and subsequently adhering to a defined policy. Through its capacity to iteratively refine its understanding of state-action values, DQN transcends traditional reinforcement learning approaches, offering unparalleled adaptability and efficiency in navigating complex decision-making scenarios.

The working process of Deep-Q network can be summarized as follows:

- **State Representation:** Convert the current state of the environment into a suitable numerical representation, such as raw pixel values or preprocessed features.
- **Neural Network Architecture:** Design a deep neural network, typically a convolutional neural network (CNN), that takes the state as input and outputs action-values for each possible action.
- **Experience Replay:** Store the agent's experiences consisting of state, action, reward, and next state tuples in a replay memory buffer.
- **Q-Learning Update:** Sample mini-batches of experiences from the replay memory to update the neural network weights. The update is performed using the loss function derived from the Bellman equation, which minimizes the discrepancy between the predicted and target action-values.

- **Exploration and Exploitation:** Balance exploration and exploitation by selecting actions either greedily based on the current policy or stochastically to encourage exploration.
- **Target Network:** Use a separate target network with the same architecture as the main network to stabilize the learning process. Periodically update the target network by copying the weights from the main network.
- **Repeat Steps 1 to 6:** Interact with the environment, gather experiences, update the network, and refine the policy iteratively until convergence.

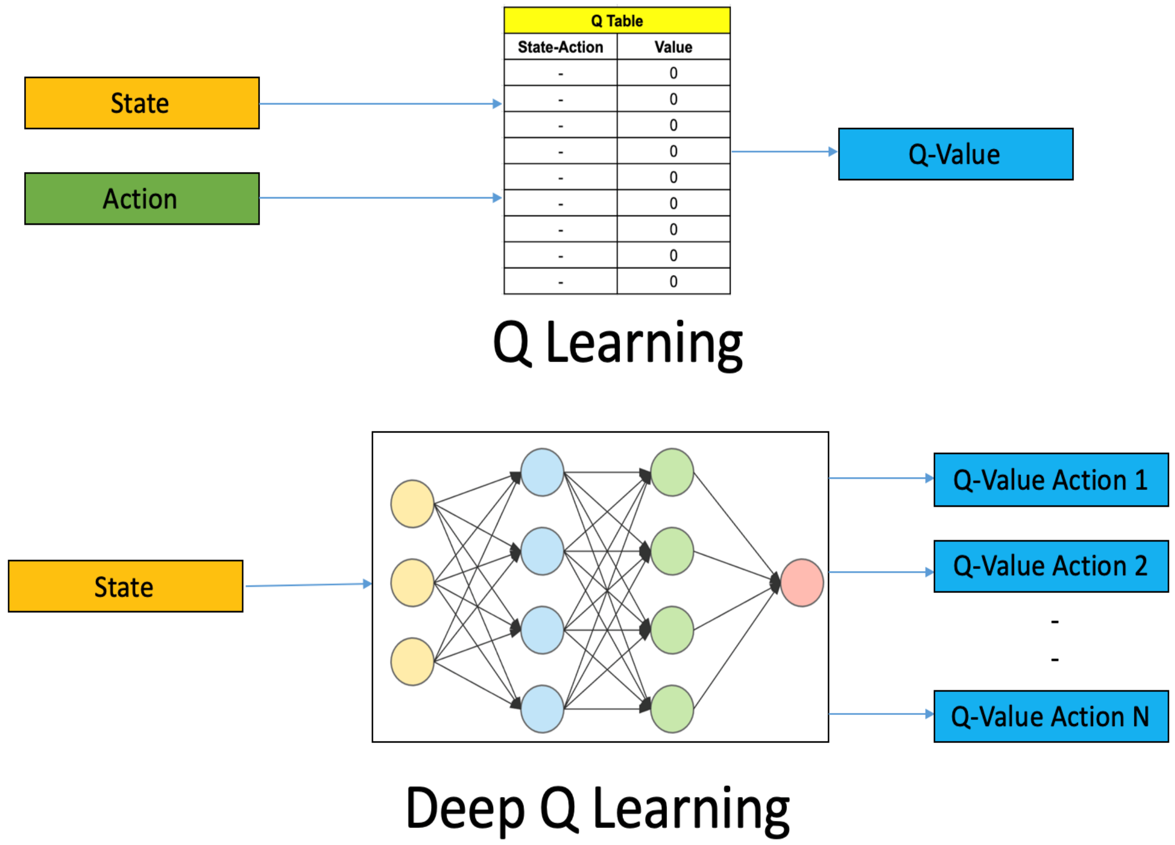


Figure.2 Deep Q Learning architecture

The Q-value for a state-action pair (s, a) is represented by Q(s, a), and it is updated iteratively to converge towards the optimal Q-values. The update rule for Q(s, a) in Q-learning is as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right]$$

where alpha is the learning rate or step size. This simply determines to what extent newly acquired information overrides old information.

By employing DQN for classification tasks, you can leverage its ability to learn complex decision-making policies and make predictions based on learned Q-values, potentially achieving high accuracy in image classification.

6.2. Actor-Critic Method:

The actor-critic algorithm represents a sophisticated fusion of policy-based and value-based reinforcement learning methodologies, aiming to harness the strengths of each while mitigating their individual limitations. Within this framework, the actor assumes the responsibility of decision-making, employing a learned policy to select actions based on the current state of the environment. Concurrently, the critic serves as an evaluator, tasked with assessing the quality or value of the actions undertaken by the actor.

This hybrid setup facilitates a dynamic interplay between exploration and exploitation, a crucial aspect of effective reinforcement learning. By leveraging the actor's ability to make informed decisions guided by learned policies and the critic's capacity to provide valuable feedback through action evaluation, the algorithm strikes a delicate balance conducive to optimal learning and performance. This symbiotic relationship between the actor and the critic empowers the algorithm to navigate complex decision spaces with agility and efficiency, making it a versatile and potent tool in various real-world applications.

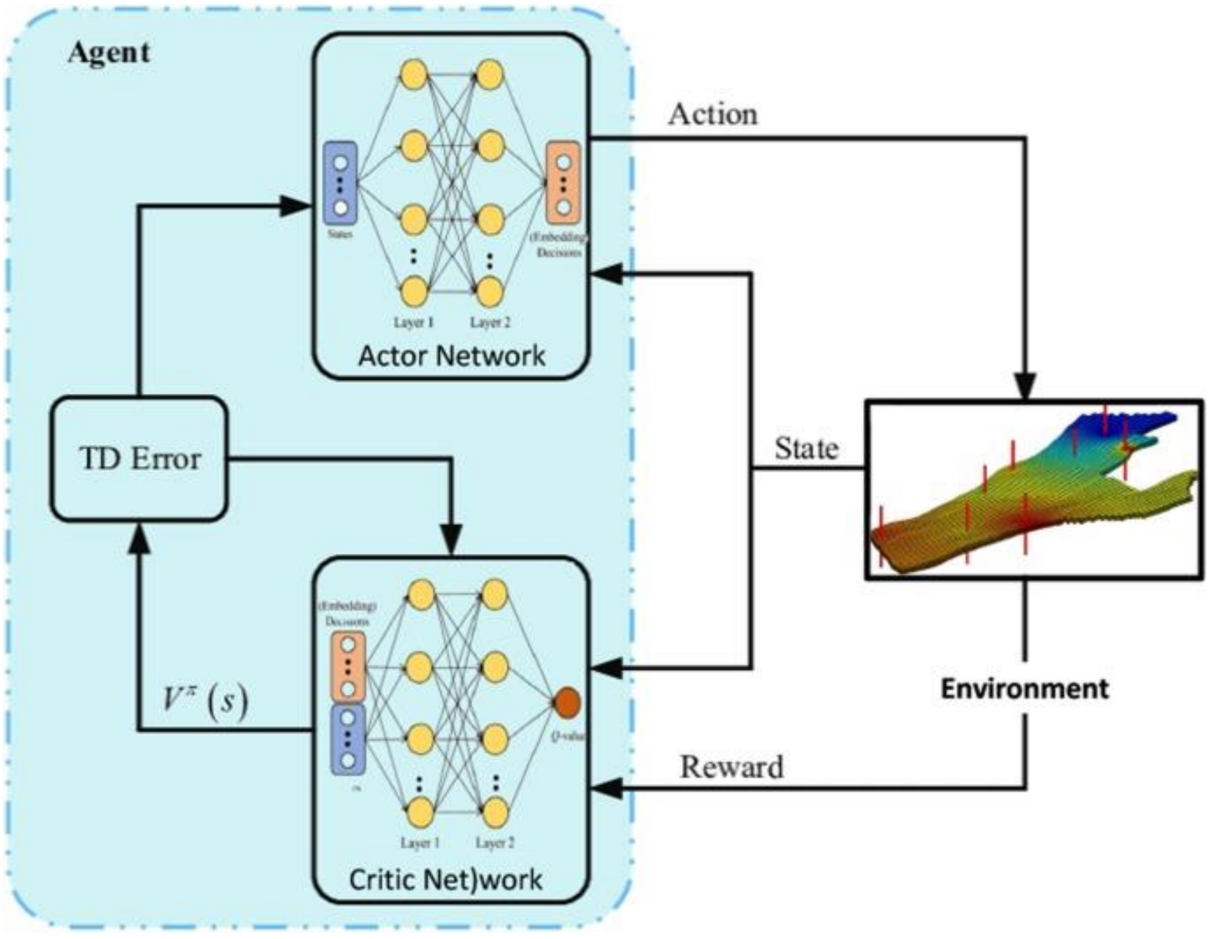


Figure.3 Actor Critic method architecture

Roles of Actor and Critic-

Actor: The actor makes decisions by selecting actions based on the current policy. Its responsibility lies in exploring the action space to maximize expected cumulative rewards. By continuously refining the policy, the actor adapts to the dynamic nature of the environment.

Critic: The critic evaluates the actions taken by the actor. It estimates the value or quality of these actions by providing feedback on their performance. The critic's role is pivotal in guiding the actor towards actions that lead to higher expected returns, contributing to the overall improvement of the learning process

Working of Actor-Critic method:

1. Take sample {s_t, a_t} using the policy πθ from the actor-network.

2. Evaluate the advantage function A_t. It can be called as TD error δt. In Actor-critic algorithm, advantage function is produced by the critic network.

$$A_{\pi_\theta}(s_t, a_t) = r(s_t, a_t) + V_{\pi_\theta}(s_{t+1}) - V_{\pi_\theta}(s_t)$$

3. Evaluate the gradient using the below expression:

$$\nabla J(\theta) \approx \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t, s_t) A_{\pi_\theta}(s_t, a_t)$$

4. Update the policy parameters, θ

$$\theta = \theta + \alpha \nabla J(\theta)$$

5. Update the weights of the critic based value-based RL(Q-learning). δt is equivalent to advantage function

$$w = w + \alpha \delta_t$$

6. Repeat 1 to 5 until we find the optimal policy πθ.

7. CONCLUSION AND FUTURE WORK

In conclusion, the automated classification of retinal diseases using fundus images represents significant advancement in the field of ophthalmology with profound implications for early diagnosis and intervention. This literature survey has highlighted the diverse methodologies and notable advancements made by researchers in this domain, ranging from deep learning approaches to feature engineering and multimodal data integration.

The utilization of convolutional neural networks (CNNs) has demonstrated remarkable performance in extracting relevant features from raw fundus images, enabling accurate classification of cataract severity levels. Additionally, traditional machine learning techniques coupled with handcrafted feature engineering methods have shown promise, particularly when combined with domain expertise.

The model has achieved the accuracy of 79.56%. Furthermore, the integration of multimodal data sources, such as optical coherence tomography (OCT) and visual acuity measurements, holds potential for enhancing classification accuracy and capturing a more comprehensive understanding of cataract progression.

Despite these advancements, challenges such as limited annotated datasets and the need for interpretability and explainability in clinical settings remain. Collaborative efforts between researchers, clinicians, and industry partners are crucial for addressing these challenges and advancing the development of reliable automated screening tools for cataracts.

Moving forward, several avenues of future work emerge from the current state of research in automated classification of cataracts and other eye diseases using fundus images:

Dataset Expansion and Standardization: Future research should focus on the creation of larger and more diverse annotated datasets encompassing various cataract severity levels, different patient demographics, and imaging conditions. Standardization of data collection protocols and annotation methodologies will be essential to ensure dataset consistency and facilitate model generalization.

Advanced Deep Learning Architectures: Exploration of novel deep learning architectures tailored specifically for fundus image analysis could lead to further improvements in classification accuracy and robustness. Architectures capable of leveraging hierarchical information across multiple scales and incorporating attention mechanisms to focus on salient features could enhance model performance.

Interpretability and Explainability: Enhancing the interpretability and explainability of automated classification models is crucial for gaining trust and acceptance in clinical practice. Future research should focus on developing techniques to provide clinicians with insights into the decision-making process of these models, including visualization of important features and factors influencing classification outcomes.

Clinical Validation and Integration: Rigorous validation of automated classification models in real-world clinical settings is necessary to assess their performance, reliability, and impact on patient outcomes. Integration of these models into existing clinical workflows, alongside validation by ophthalmologists and healthcare professionals, will be essential to demonstrate their effectiveness and utility in improving diagnostic accuracy and patient care.

Multimodal Fusion and Integration: Further exploration of multimodal data fusion and integration techniques, including combining fundus images with additional imaging modalities such as OCT and visual acuity measurements, could lead to more comprehensive and holistic assessments of eye health. Developing effective fusion strategies to leverage complementary information from different modalities could enhance classification accuracy and provide a more nuanced understanding of eye diseases.

Longitudinal Analysis and Disease Progression Modeling: Investigating longitudinal changes in fundus images and developing models to predict disease progression could enable proactive management and personalized treatment strategies for patients at risk of developing cataracts and other eye diseases. Longitudinal analysis techniques, coupled with advanced machine learning and deep learning methodologies, could offer valuable insights into disease dynamics and treatment response over time.

Overall, future research endeavors in automated classification of cataracts and other eye diseases should aim to address existing challenges, leverage emerging technologies, and prioritize clinical validation to translate advancements into tangible benefits for patients and healthcare providers. By embracing interdisciplinary collaboration and innovation, the field holds immense potential to revolutionize early diagnosis, intervention, and management of ocular conditions, ultimately improving visual outcomes and quality of life for individuals worldwide.

REFERENCES

- Zhiqiang Qiao , Qinyan Zhang , Yanyan Dong ,and Ji-Jiang Yang, ‘Application of SVM Based on Genetic Algorithm in Classification of Cataract Fundus Images’, 2017 IEEE International Conference on Imaging Systems and Techniques (IST), **DOI:**10.1109/IST.2017.8261541
- V Harini and V Bhanumathi, ‘Automatic Cataract Classification System’, 2016 International Conference on Communication and Signal Processing (ICCSP), **DOI:** 10.1109/ICCSP.2016.7754258
- Ri Munarto, Mochtar Ali Setyo Yudono, and Endi Permata, ‘Automatic Cataract Classification System Using Neural Network Algorithm Backpropagation ’, 2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE), **DOI:** 10.1109/ICIEE49813.2020.9277441
- Turimerla Pratap and Priyanka Kokil, ‘Automatic Cataract Detection in Fundus Retinal Images using Singular Value Decomposition’, 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), **DOI:** 10.1109/WiSPNET45539.2019.9032867
- Kaustubh Funde, Jai Joshi, Karen Castelino, Pranjal Patil, and Nikahat Mulla, ‘Cataract Detection by Leveraging VGG-19 Classification Model on Retinal Images’, 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), **DOI:** 10.1109/ICCCNT54827.2022.9984433
- Kashish Chauhan, Rajesh Kumar Yadav, Kashish, and Kartik Dagar, ‘Cataract detection from eye fundus image using an ensemble of transfer learning models’, 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), DOI: 10.1109/ICACITE53722.2022.9823638
- Raghavendra Chaudhary and Arun Kumar, ‘Cataract Detection using Deep Learning Model on Digital Camera Images’, 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), DOI: 10.1109/CYBERNETICSCOM55287.2022.9865591
- MASUM SHAH JUNAYED, MD BAHARUL ISLAM, (Senior Member, IEEE), AREZOO SADEGHZADEH , AND SAIMUNUR RAHMAN ‘CataractNet: An Automated Cataract Detection System Using Deep Learning for Fundus Images’, IEEE Access (Volume: 9), **DOI:** 10.1109/ACCESS.2021.3112938

- Turimerla Pratap and Priyanka Kokil, ‘Computer-aided Cataract Grading Under Adversarial Environment’, 2022 IEEE International Conference on Signal Processing and Communications (SPCOM), DOI:10.1109/SPCOM55316.2022.9840821
- Gunjan Sharma, Vatsala Anand, and Sheifali Gupta, ‘Cracking Light on Cataract Detection by Implementing VGG16 Transfer Learning-Based Model on Fundus Images ’, 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), DOI: 10.1109/RMKMATE59243.2023.10368860
- Naitik Vora , Vighnesh Iyer , and Harshal Dalvi, ‘Early Diagnosis of Cataract and Diabetic Retinopathy for Rural India using a Cloud-Based Deep Learning Model ’, 2023 International Conference on Advanced Computing Technologies and Applications (ICACTA), DOI: 10.1109/ICACTA58201.2023.10392985
- Jin Zheng, Liye Guo, Lihui Peng, Jianqiang Li, Jijiang Yang, and Qingfeng Liang, ‘Fundus Image Based Cataract Classification’, 2014 IEEE International Conference on Imaging Systems and Techniques (IST) Proceedings, **DOI:** 10.1109/IST.2014.6958452
- **https://medium.com/@shruti.dhumne/deep-q-network-dqn-90e1a8799871**
- **https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/**
- **https://medium.com/intro-to-artificial-intelligence/the-actor-critic-reinforcement-learning-algorithm-c8095a655c14**
- **https://www.geeksforgeeks.org/actor-critic-algorithm-in-reinforcement-learning/**

APPENDIX

SOURCE CODE

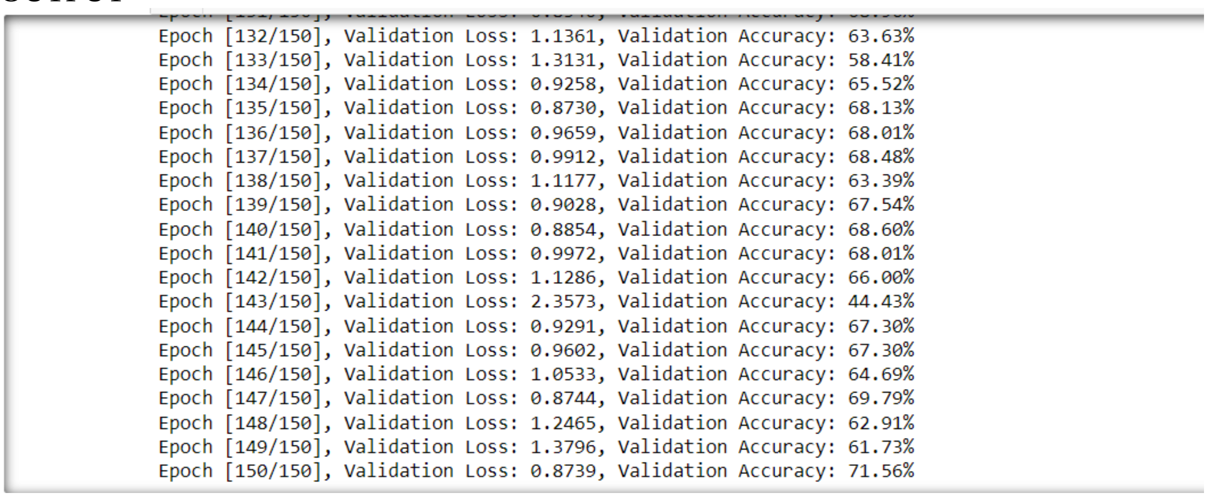
TRAINING:-

```
import torch
import pickle
import torch.nn as nn
import torch.optim as optim
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from torch.utils.data import DataLoader, Dataset

# Define the DQN model
class DQN(nn.Module):
    def __init__(self, input_size, hidden_size1, hidden_size2, hidden_size3, hidden_size4,
num_classes):
        super(DQN, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size1)
        self.bn1 = nn.BatchNorm1d(hidden_size1)
        self.relu1 = nn.ReLU()
        self.fc2 = nn.Linear(hidden_size1, hidden_size2)
        self.bn2 = nn.BatchNorm1d(hidden_size2)
        self.relu2 = nn.ReLU()
        self.fc3 = nn.Linear(hidden_size2, hidden_size3)
        self.bn3 = nn.BatchNorm1d(hidden_size3)
        self.relu3 = nn.ReLU()
        self.fc4 = nn.Linear(hidden_size3, hidden_size4)
        self.bn4 = nn.BatchNorm1d(hidden_size4)
        self.relu4 = nn.ReLU()
        self.fc5 = nn.Linear(hidden_size4, num_classes)

    def forward(self, x):
        x = self.fc1(x)
        x = self.bn1(x)
        x = self.relu1(x)
        x = self.fc2(x)
        x = self.bn2(x)
        x = self.relu2(x)
        x = self.fc3(x)
        x = self.bn3(x)
        x = self.relu3(x)
        x = self.fc4(x)
        x = self.bn4(x)
        x = self.relu4(x)
        x = self.fc5(x)
        return x
```

```
# Load the data from CSV
data = pd.read_csv(r"C:\Users\Naeem\OneDrive\Documents\Train_line_averages_4000.csv")

# Encode the target labels
label_encoder = LabelEncoder()
data['TYPE'] = label_encoder.fit_transform(data['TYPE'])

# Split data into features and labels
X = data.iloc[:, 1:].values
y = data.iloc[:, 0].values

# Split data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Convert data to PyTorch tensors
X_train_tensor = torch.tensor(X_train, dtype=torch.float32)
y_train_tensor = torch.tensor(y_train, dtype=torch.long)
X_val_tensor = torch.tensor(X_val, dtype=torch.float32)
y_val_tensor = torch.tensor(y_val, dtype=torch.long)

# Define model parameters
input_size = X_train.shape[1]
num_classes = len(np.unique(y_train))

# Instantiate the model
model = DQN(input_size, 256, 128, 64, 32, num_classes)

# Define loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# Define a PyTorch Dataset for DataLoader
class CustomDataset(Dataset):
    def __init__(self, features, labels):
        self.features = features
        self.labels = labels

    def __len__(self):
        return len(self.features)

    def __getitem__(self, index):
        return self.features[index], self.labels[index]

# Create DataLoader for training and validation sets
train_dataset = CustomDataset(X_train_tensor, y_train_tensor)
val_dataset = CustomDataset(X_val_tensor, y_val_tensor)
train_loader = DataLoader(train_dataset, batch_size=64, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=64)
```

```
# Training the model
num_epochs = 150
best_val_loss = float('inf')  # Track the best validation loss
for epoch in range(num_epochs):
    model.train()
    for features, labels in train_loader:
        optimizer.zero_grad()
        outputs = model(features)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

    model.eval()
    with torch.no_grad():
        val_loss = 0.0
        total = 0
        correct = 0
        for features, labels in val_loader:
            outputs = model(features)
            _, predicted = torch.max(outputs, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()
            val_loss += criterion(outputs, labels).item()

    # Save the model weights if the validation loss has decreased
    if val_loss < best_val_loss:
        best_val_loss = val_loss
        # Serialize model weights and save to a .pkl file
        with open('best_model.pkl', 'wb') as f:
            pickle.dump(model.state_dict(), f)

    print(f"Epoch [{epoch + 1}/{num_epochs}], "
          f"Validation Loss: {val_loss / len(val_loader):.4f}, "
          f"Validation Accuracy: {(correct / total) * 100:.2f}%")
```

OUTPUT

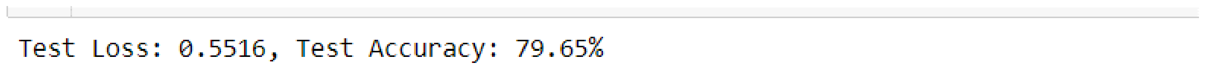


```
Epoch [132/150], Validation Loss: 1.1361, Validation Accuracy: 63.63%
Epoch [133/150], Validation Loss: 1.3131, Validation Accuracy: 58.41%
Epoch [134/150], Validation Loss: 0.9258, Validation Accuracy: 65.52%
Epoch [135/150], Validation Loss: 0.8730, Validation Accuracy: 68.13%
Epoch [136/150], Validation Loss: 0.9659, Validation Accuracy: 68.01%
Epoch [137/150], Validation Loss: 0.9912, Validation Accuracy: 68.48%
Epoch [138/150], Validation Loss: 1.1177, Validation Accuracy: 63.39%
Epoch [139/150], Validation Loss: 0.9028, Validation Accuracy: 67.54%
Epoch [140/150], Validation Loss: 0.8854, Validation Accuracy: 68.60%
Epoch [141/150], Validation Loss: 0.9972, Validation Accuracy: 68.01%
Epoch [142/150], Validation Loss: 1.1286, Validation Accuracy: 66.00%
Epoch [143/150], Validation Loss: 2.3573, Validation Accuracy: 44.43%
Epoch [144/150], Validation Loss: 0.9291, Validation Accuracy: 67.30%
Epoch [145/150], Validation Loss: 0.9602, Validation Accuracy: 67.30%
Epoch [146/150], Validation Loss: 1.0533, Validation Accuracy: 64.69%
Epoch [147/150], Validation Loss: 0.8744, Validation Accuracy: 69.79%
Epoch [148/150], Validation Loss: 1.2465, Validation Accuracy: 62.91%
Epoch [149/150], Validation Loss: 1.3796, Validation Accuracy: 61.73%
Epoch [150/150], Validation Loss: 0.8739, Validation Accuracy: 71.56%
```

TESTING:-

```
import pandas as pd
import torch
from sklearn.preprocessing import LabelEncoder
from torch.utils.data import DataLoader, Dataset

# Load the test data from CSV
test_data =
pd.read_csv(r"C:\Users\Naeem\OneDrive\Documents\Train_line_averages_4000.csv")

# Encode the target labels
label_encoder = LabelEncoder()
test_data['TYPE'] = label_encoder.fit_transform(test_data['TYPE'])

# Split test data into features and labels
X_test = test_data.iloc[:, 1:].values
y_test = test_data.iloc[:, 0].values

# Convert test data to PyTorch tensors
X_test_tensor = torch.tensor(X_test, dtype=torch.float32)
y_test_tensor = torch.tensor(y_test, dtype=torch.long)

# Define a PyTorch Dataset for DataLoader
class CustomDataset(Dataset):
    def __init__(self, features, labels):
        self.features = features
        self.labels = labels

    def __len__(self):
        return len(self.features)

    def __getitem__(self, index):
        return self.features[index], self.labels[index]

# Create DataLoader for test set
test_dataset = CustomDataset(X_test_tensor, y_test_tensor)
test_loader = DataLoader(test_dataset, batch_size=64)

# Evaluate the model on test data
model.eval()
with torch.no_grad():
    test_loss = 0.0
    total = 0
    correct = 0
    for features, labels in test_loader:
        outputs = model(features)
        _, predicted = torch.max(outputs, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()
        test_loss += criterion(outputs, labels).item()

print(f"Test Loss: {test_loss / len(test_loader):.4f}, "
      f"Test Accuracy: {(correct / total) * 100:.2f}%")
```

OUTPUT:-

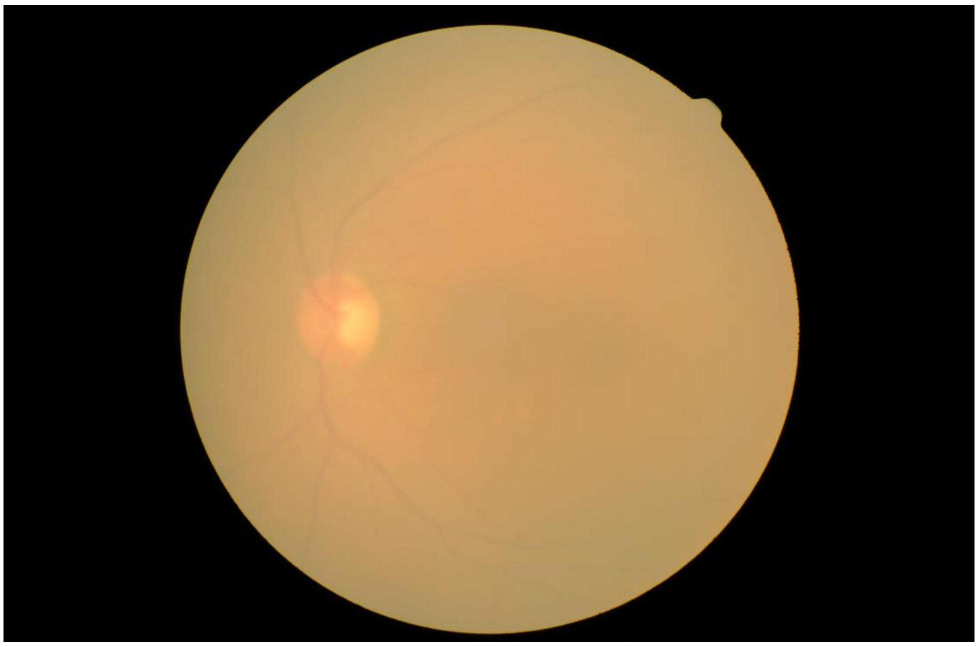


```
Test Loss: 0.5516, Test Accuracy: 79.65%
```

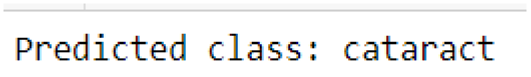
<u>CLASSIFICATION:-</u>

```
import torch
from PIL import Image
import numpy as np

# Define a function to preprocess the input image
def preprocess_image(image_path):
    # Open the image file
    image = Image.open(image_path)

    # Resize the image to match the input size of the model
    resized_image = image.resize((512, 512))

    # Convert the image to grayscale
    grayscale_image = resized_image.convert("L")

    # Convert the grayscale image to a NumPy array
    pixel_array = np.array(grayscale_image)

    # Compute the average of each line
    line_averages = np.mean(pixel_array, axis=1)

    # Convert the line averages to a PyTorch tensor
    input_tensor = torch.tensor(line_averages, dtype=torch.float32)

    # Reshape the input tensor to match the expected input size of the model
    input_tensor = input_tensor.view(1, -1)

    return input_tensor

# Define a function to predict the class of the input image
def predict_image_class(model, input_image_tensor):
    # Set the model to evaluation mode
    model.eval()

    # Perform prediction
    with torch.no_grad():
        output = model(input_image_tensor)
        _, predicted_class = torch.max(output, 1)
    # Convert the predicted class tensor to an integer
    predicted_class = predicted_class.item()

    return predicted_class

# Load the trained model
# model = torch.load("batch_normalisation.pth")

# Preprocess the input image
input_image_path = r"C:\Users\Naeem\Downloads\dataset\cataract\cataract_092.png"
input_image_tensor = preprocess_image(input_image_path)

# Predict the class of the input image
predicted_class = predict_image_class(model, input_image_tensor)

# Decode the predicted class using the label encoder used during training
predicted_class_name = label_encoder.inverse_transform([predicted_class])[0]

print("Predicted class:", predicted_class_name)
```

INPUT:-



OUTPUT:-



```
Predicted class: cataract
```