

BRAIN STROKE ANALYSIS FROM NON-CONTRAST BRAIN CT AND PATH-PLANNING FOR ROBOT-ASSISTED THROMBECTOMY

A PROJECT REPORT

Submitted By

A ANIRUDH 195001015

KARTHIK D 195001047

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



Department of Computer Science and Engineering

**Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)**

Kalavakkam - 603110

May 2023

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled "**"BRAIN STROKE ANALYSIS FROM NON-CONTRAST BRAIN CT AND PATH-PLANNING FOR ROBOT-ASSISTED THROMBECTOMY"**" is the *bonafide* work of "**A ANIRUDH (195001015) and KARTHIK D (195001047)**" who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. T. T. MIRNALINEE
HEAD OF THE DEPARTMENT

Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Dr. T. T. MIRNALINEE
SUPERVISOR

Professor and Head,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Place:

Date:

Submitted for the examination held on.....

Internal Examiner

External Examiner

ACKNOWLEDGEMENTS

I thank GOD, the almighty for giving me the strength and knowledge to do this project.

I would like to thank and deep sense of gratitude to my guide **Dr. T. T. MIRNALINEE**, Professor and Head, Department of Computer Science and Engineering, for her valuable advice and suggestions as well as her continued guidance, patience and support that helped me to shape and refine my work.

My sincere thanks to **Dr. T. T. MIRNALINEE**, Professor and Head of the Department of Computer Science and Engineering, for her words of advice and encouragement and I would like to thank our project Coordinator **Dr. B. BHARATHI**, Associate Professor, Department of Computer Science and Engineering for her valuable suggestions throughout this project.

I express my deep respect to the founder **Dr. SHIV NADAR**, Chairman, SSN Institutions. I also express my appreciation to our **Dr. V. E. ANNAMALAI**, Principal, for all the help he has rendered during this course of study.

I would like to extend my sincere thanks to all the teaching and non-teaching staffs of our department who have contributed directly and indirectly during the course of my project work. Finally, I would like to thank my parents and friends for their patience, cooperation, and moral support throughout my life.

A Anirudh

Karthik D

ABSTRACT

Brain stroke is among the most common reason for death around the world [25]. Brain imaging methods like magnetic resonance imaging (MRI), diffuse optical imaging, functional MRIs, and CT are quite helpful for initial screening. However, sophisticated imaging techniques like MRI require high operating cost and a well-trained operator. Often times, CT is the most convenient and quick imaging modality. Toward the diagnosis of strokes, physical and manual examination of the patient is performed to determine an appropriate course of treatment, aided by brain images. This process is made complex by the contrasting and divergent treatments required for the two kinds of stroke – ischemic and hemorrhagic – requiring accurate lesion type identification. Furthermore, an accurate segmentation of the infarct location is particularly useful for subsequent treatment. However, manually performing this diagnosis is extremely time-consuming for the already overburdened radiologist, and is prone to human-error. Thus, there is a need for an effective automated system for diagnosing strokes that can help physicians quickly start treatment after stroke onset. Hence, in this work, we proposed to develop a deep learning -based computation pipeline to process CT slices to detect stroke type and localize infarct location. Furthermore, we propose to reconstruct the CT slices into volumetric models while correcting errors in infarct localization, and potentially draw out diagnostic measurements that could help with subsequent treatment. Finally, we also propose a path planning algorithm to treat strokes using a six degree of freedom (DOF) arm. This Q-learning based method is simulated qualitatively on benchmark virtual environments.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
1 INTRODUCTION	1
1.1 MOTIVATION	1
1.2 BACKGROUND	2
1.3 PROBLEM DEFINITION	4
1.4 ORGANIZATION OF THE REPORT	6
2 LITERATURE SURVEY	7
2.1 NCCT ANALYSIS TO LOCALIZE STROKE REGIONS	7
2.2 PATH PLANNING FOR ROBOTIC MANIPULATORS	12
2.3 GAPS IDENTIFIED	13
2.4 RESEARCH OBJECTIVES	15
3 PROPOSED METHODOLOGY	17
3.1 DETECTION AND LOCALIZATION OF HEMORRHAGIC STROKE REGIONS	19
3.1.1 Data Collection	20
3.1.2 Slice-level Annotation to Establish Ground Truth	21
3.1.3 Data Pre-processing	22

3.1.4	Detection and Segmentation Model Building	24
3.1.5	Visualization of Hemorrhagic Regions on NCCT Scans . .	26
3.1.6	Diagnostic Inference from Volumes	27
3.1.7	Validation from Radiologists	27
3.2	ROBOTIC PATH PLANNING MODULE	28
3.2.1	Deep Reinforcement Learning	29
3.2.2	Q-Learning	32
3.2.3	Enhancements to Q-Learning	35
3.3	PROOFS OF HYPOTHESES USED	38
3.3.1	On The Effectiveness Of Slice-wise Segmentation	38
3.3.2	On The Convergence Of A Novel Distance Metric For Path Planning In Environments With Obstacles	46
4	EXPERIMENTAL RESULTS	51
4.1	DATASET AND ENVIRONMENT DESCRIPTION	51
4.1.1	Dataset	51
4.1.2	Environments for path-planning	52
4.2	ECOSYSTEM	52
4.2.1	Hardware	52
4.2.2	Software	53
4.3	EXPERIMENTS CONDUCTED	53
4.4	VISUALIZATION	54
5	PERFORMANCE ANALYSIS	56
5.1	NCCT SCAN ANALYSIS FOR STROKE REGION LOCALIZATION	56

5.2	PATH PLANNING FOR SURGICAL INTERVENTION	59
6	SOCIAL IMPACT AND SUSTAINABILITY	61
7	CONCLUSIONS AND FUTURE WORK	64
7.1	CONCLUSION	64
7.2	FUTURE WORK	65
	REFERENCES	65
A	Q-LEARNING AND ROBOTICS TECHNIQUES	73
A.1	INVERSE KINEMATICS	73
A.2	OPENAI GYM	74
A.3	ROBOTICS, VISION & CONTROL TOOLBOX	74

LIST OF TABLES

3.1	Experimental parameters for DDPG and TD3.	32
3.2	Results for DDPG and TD3 experiments.	32
5.1	Comparison of segmentation approaches on open-access datasets.	58

LIST OF FIGURES

1.1	Samples of CT slices to depict the infarcts arising from <i>Ischemic</i> and <i>Hemorrhagic</i> strokes.	5
2.1	Samples of CT slices to depict <i>overlapping infarcts</i> – infarcts of multiple types at the same image region.	11
3.1	Architecture of the proposed solution.	19
3.2	Overview of the NCCT analysis workflows.	20
3.3	An unsupervised approach to segment stroke regions from brain CTs.	22
3.4	Pre-processing methods to apply on the NCCT slices before passing them to the neural network and computational analysis pipeline to correct artifacts.	23
3.5	Proposed network architecture for scan segmentation.	26
3.6	The PandaReach-v2 robotic environment.	31
3.7	Path planning workflow based on Q-learning for the 6DOF robotic arm.	34
4.1	Visualization of NCCT slices and segmentation results for clinicians to identify the stroke location using the web application.	54
4.2	Visualization of path-planning for surgical intervention using the web application.	55
5.1	Qualitative comparison of segmentation methods.	57
5.2	Intermediate states of the 6-DOF robotic arm when implementing the planned path in one of the seven successful environments successfully planned by the proposed algorithm.	60

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The second most common cause of mortality worldwide, behind heart disease, is brain stroke [28]. The majority of survivors must deal with permanent or ongoing injury. A doctor might begin the initial screening of the patient using brain imaging techniques including magnetic resonance imaging (MRI) and computed tomography (CT). There are numerous imaging techniques available for studying the brain, such as positron emission tomography, functional MRI, magnetoencephalography, diffuse optical imaging, and X-ray imaging [6, 51]. However, each of these imaging methods has a high operating cost and requires a skilled operator, so not all clinics and hospitals may have access to them.

In this study, a method for automatically classifying brain CT scan images has been proposed. This method predicts the category to which a brain CT scan image belongs, as well as the locations of stroke infarcts for hemorrhagic and ischemic stroke types, and normal tissue. *Hemorrhagic stroke* happens when a weak blood artery bursts and leaks into the surrounding brain tissues. On the other hand, a *ischemic stroke* happens when a blood clot prevents blood flow to the brain. According to [23], brain haemorrhage can also happen after an ischemic stroke and is a very dangerous consequence. Figure 1.1 shows sample images of the two forms of stroke on CT slices, with infarct locations highlighted in red.

1.2 BACKGROUND

In medical imaging, image classification is frequently utilised [37]. However, the classification system's results should be as accurate as a manual diagnosis for best results. Because it automatically calculates features within the convolutional layers of the deep system [2, 29], deep learning is now widely used as a classification method. The key benefit of employing deep learning is that it performs better than other traditional approaches for classifying images, according to [44]. There are numerous deep learning techniques that have been developed, including recurrent neural networks, long short-term memory (LSTM) [17], CNNs [35], deep belief nets (DBN) [16], and others. CNN has been used frequently among these techniques for computer vision and medical image processing issues including ImageNet, face recognition, digit classification of home numbers, patch classification from medical pictures, etc. Random forest (RF), nearest neighbours (NN), decision tree (DT), multilayer perceptron (MLP), support vector machine (SVM), and many more classification techniques are alternatives to deep learning techniques [43, 45, 47].

Towards the diagnosis of stroke, doctors first confirm symptoms from the patient or family members. The most important thing which will help in accurate identification is the stroke history. This is usually followed by neuroimaging of the patient. The most commonly used neuroimaging method used by experts is CT scan for diagnosing brain strokes, on account of their greater affordability. The prediction of stroke from CT scan images serve as the initial step towards the proper diagnosis of a patient. These images are then sent to a cardiovascular radiologist to identify the stroke type. Thereafter, physical and manual examination of the patient is performed to determine an appropriate course of

treatment. However, this manual diagnosis procedure is extremely time-consuming for the already overburdened radiologist, and is prone to human-error. Many researchers have worked in this area for providing a computer-aided diagnosis (CAD) system, in spite of the significant requirement, there is no clinically accepted CAD system for stroke [32, 36]. The system proposed by [32] was not automated and requires significant performance improvements to warrant automation. Peixoto and Filho [36] have worked on the classification of hemorrhagic and ischemic stroke CT scan images. Their method was developed for small dataset where each case had 100 images and for future work they have to test their method on large datasets. Thus, there is a need for an effective automated system for diagnosing strokes that can help physicians quickly start treatment after stroke onset. Thus, in this work, a deep learning-based computation pipeline was proposed to process CT slices for detecting stroke type and localizing infarct location. Additionally, it is proposed to reconstruct the CT slices into volumetric models, while correcting errors in infarct localization. The resulting volume can be transformed into a computationally pliable 3D model, which can be utilized for diagnostic measurements and treatment simulations.

Doctors first verify the patient’s or the patient’s family members’ reports of symptoms before making the stroke diagnosis. The history of strokes is the most crucial factor that will aid in proper identification. The patient’s neuroimaging often comes next. Because they are more accessible and more frequently used by professionals, CT scans are the most popular neuroimaging technique for identifying brain strokes. The first step towards a patient’s correct diagnosis is the stroke prediction using CT scan data. The type of stroke is subsequently determined by a cardiovascular radiologist using these images. The patient is then

subjected to a physical and manual examination in order to establish the best course of treatment. For the already overworked radiologist, this manual diagnosis process is incredibly time-consuming and prone to human mistake. Despite the fact that numerous academics have worked on developing a computer-aided diagnosis (CAD) system, there is still no clinically acceptable CAD system for stroke [32, 36]. The approach suggested in [32] was not automated, and automation would only be justified with considerable performance gains. Research has been done on categorising CT scan pictures of hemorrhagic and ischemic strokes [36]. They had to test their strategy on large datasets because it was built for a tiny dataset with only 100 photos in each example. As a result, a reliable automated system is required for diagnosing strokes so that doctors can rapidly begin treating patients after a stroke occurs. As a result, a deep learning-based computing pipeline was presented in this work to process CT slices for stroke type detection and infarct location localization. It is also suggested to reconstruct the CT slices into volumetric models while making infarct localization corrections. The generated volume can be converted into a 3D model that is computationally flexible and can be used for measurements during diagnosis and simulations during treatment.

1.3 PROBLEM DEFINITION

This research proposes to achieve the following objectives through data-driven computational modeling and analysis. The approach and methods that will be adopted in meeting these objectives is detailed in the subsequent section. The specific objectives sought to be achieved in this project are as follows:

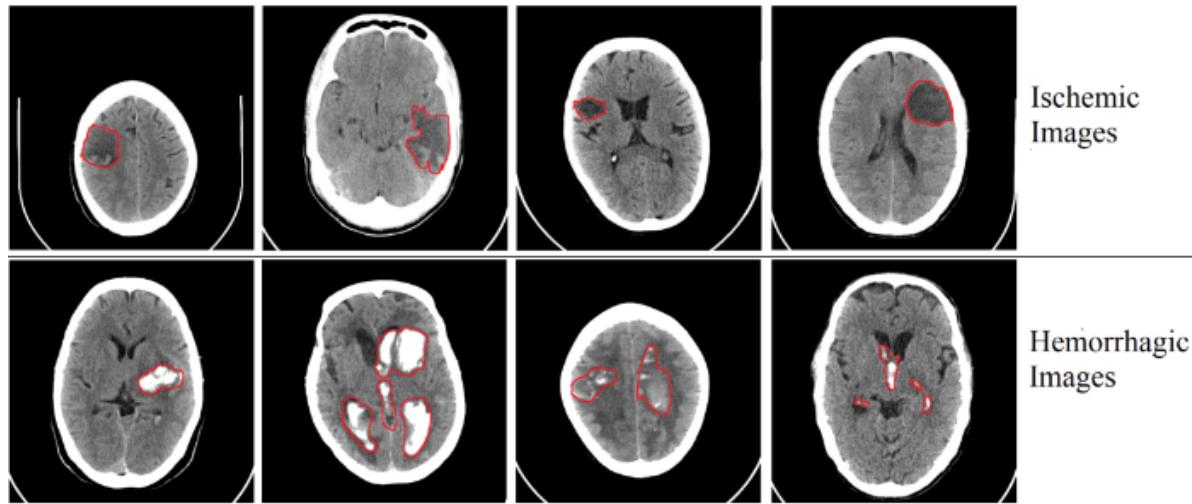


FIGURE 1.1: Samples of CT slices to depict the infarcts arising from *Ischemic* and *Hemorrhagic* strokes.

- To detect the presence of, localize, and segment out stroke regions from non-contrast computed tomography (NCCT) volumes of the brain.
- To perform context-aware segmentation by fusing multi-dimensional features when processing scan volumes to improve overall stroke region localization performance by exploiting geometric constraints.
- To infer location information and other quantitative diagnostic measures from the reconstructed volumes to assess the condition such as cerebral infarct volume (CIT), when associated metadata relating slice dimensions to real-word dimensions.
- To devise a path-planning strategy to perform surgical intervention using a six degree-of-freedom robotic arm.
- To develop a web-based end-user application that can be used by clinicians to digitally browse scans and visualize predicted stroke regions to assist them with diagnosis and with devising treatment plans.

- To allow clinicians to visualize the planned-path for surgical intervention in the simulation environment through the web-based application.

1.4 ORGANIZATION OF THE REPORT

The report is organized as follows:

Chapter 1 provides a brief introduction to the project, including the motivation behind it and the problem statement that is being addressed. *Chapter 2* presents a literature survey that discusses existing work related to the problem statement and identifies gaps in the field. The research objectives of the project are also presented in this chapter. *Chapter 3* describes the proposed methodology for addressing the problem statement. The chapter is divided into three sections, each detailing a specific module of the proposed system. The first section focuses on the detection and localization of hemorrhagic stroke regions, the second section presents a robotic path planning module, and the third section provides proofs of the hypotheses used in the proposed methodology. *Chapter 4* presents the experimental results of the implementation and performance analysis of the proposed system. The results of the segmentation models and path planning module are discussed in this chapter. *Chapter 5* presents a detailed performance analysis of the proposed system. *Chapter 6* explores the social impact and sustainability of the project. *Chapter 7* summarizes the conclusions drawn from the project and proposes future work to be done.

CHAPTER 2

LITERATURE SURVEY

2.1 NCCT ANALYSIS TO LOCALIZE STROKE REGIONS

In general, computer-aided diagnostic (CAD) systems that analyse and analyse medical pictures can assist doctors in the early diagnosis of diseases [14, 19, 32, 49]. Here are some significant image preprocessing and classification techniques that have been used to date to analyse brain strokes from medical images.

In order to properly treat a specific injury, Saatman et al. [39] have examined numerous classification methods for traumatic brain damage. They talked about categorising brain injuries according to their severity using a variety of indicators, such as the Glasgow Coma Scale (GCS), but these processes call for several clinical trials. They also talked about pathoanatomic classification, physical mechanism classification, pathophysiology, and prognostic modelling. They assert that new technologies can be created, such as dataset development, data sharing, data mining, and bioinformatics, etc., to prevent the clinical trials of numerous patients. The Boston Acute Stroke Imaging Scale (BASIS) instrument was used in 2008 by Mozqueda et al. [46] to test the concept that acute ischemic stroke can be classified into major and mild strokes from CT or MR angiography images. They have established that the BASIS categorization tool can predict strokes more accurately than the Alberta Stroke Programme Early CT Score (ASPECTS).

An automatic technique for the detection and classification of an abnormality from low-contrast CT images into three categories — haemorrhage, chronic, and acute infarct — was presented by Chawla et al. [3]. Three steps make up their suggested method: improving the image, identifying mid-line symmetry, and classifying aberrant slices. The fundamental flaw in their work is that they only presented results based on the dataset and did not compare them to any results obtained using prior state-of-the-art techniques. They only provided the precision and recall figures for the experiments, not the accuracy they attained on their dataset. Using T1-weighted MRI scans and the Gaussian nave Bayes classification approach, Griffis et al. [13] reference griffis2016 have also worked on the identification of ischemic stroke lesions. Their approach had drawbacks in that they were unable to find very small-scale, subtle WM lesions. A method to segment and categorise haemorrhage was proposed by Shahangian and Pourghassem in [41], where segmentation was based on a modified form of distance regularised level set evolution (MDRLSE). After that, weighted grayscale histogram characteristics were extracted and utilised to categorise the photos using an SVM classifier, which had an accuracy of 94.13 percent. However, the full dataset information is not provided, including the specifics of the CT scanner and the conditions under which the images were acquired. The other drawback is that they only employed 627 photos—a fairly small number—to classify the images into the five separate groups. By employing a modified robust fuzzy c-means clustering (MRFCM) form of fuzzy clustering in combination with the distance regularisation level set evolution (DRLSE) method, Gautam et al. [10] proposed segmenting hemorrhagic stroke from CT scan pictures. Their approach, though, was unable to recognise very minute lesions from the photos. Additionally, they have suggested a system for

categorising brain CT scan images into hemorrhagic, ischemic, and normal states [9]. They have presented a brand-new feature descriptor called the local gradient of gradient pattern (LG2P) descriptor. Using fine NN and cubic SVM, they were able to get the greatest classification accuracy of 83.11% and 86.11% on a dataset of 900 images. However, this accuracy is very low and needs to be increased for better outcomes. Woo et al. [48] proposed the incremental gradient descent decision boundary feature extraction approach for the classification of the UCI dataset, where they have demonstrated results on the Parkinson speech, chess, molecular biology, gas sensor, and array drift datasets. Authors of [4, 40] present a literature review on feature extraction approaches that was recently released by multiple investigators in order to learn more ways.

Subudhi et al. [43], who used expectation-maximization and a random forest classifier, introduced a new technique for the segmentation and classification of cerebral stroke from MR images. The accuracy they attained on the collection of 192 brain pictures was 93.4 percent. The weakness of this research is the lack of description of the dataset, including how and where it was gathered. Using MR scans, Ortiz-Ramón et al. [34] suggested a texture analysis method to determine whether a patient has had a stroke or not, as well as whether that stroke is lacunar or cortical. In order to complete their work, they utilized a variety of approaches put forth by many researchers, and references to those methods may be found in [43]. They claim that the inability to merge stroke and ageing (old stroke lesion) datasets for the analysis of images displaying recent cortical or lacunar strokes is a drawback of their work. However, by enhancing the image quality, image normalization techniques can address this problem.

Deep learning approaches are receiving interest due to the reliable diagnosis

outcomes, which set them apart from the traditional machine learning methods that have been utilised extensively for the detection of diseases from medical images. In 2012, Krizhevsky et al. won the most challenging ImageNet large-scale visual recognition challenge (ILSVRC) by proposing AlexNet as the CNN architecture [22]. Later, it became known as the computer vision and machine learning revolution. Five convolutional layers, three fully connected layers, and two dropout layers with a probability of 0.5 make up AlexNet. The softmax layer can produce the best classification prediction when fed the output of the final fully connected layer. The input image size for this architecture should be $224 \times 224 \times 3$. However, many researchers have attempted to create new architectures in an effort to increase accuracy. Simonyan and Zisserman's idea for VGG16, which they first presented in 2014 [42]. Their network's primary contribution is the creation of an extremely deep CNN architecture for the recognition of very big image datasets. They did this by bringing the network's depth up to 16–19 layers and using a 3×3 convolution filter. Five batches of 13–16 convolutional layers and three fully connected layers make up the VGG16. After the maximum pooling, the number of filters increases by a factor of 2 for each layer until it reaches 512 in the first layer. The input picture size for this architecture is $224 \times 224 \times 3$, which is the same as for AlexNet. The other CNN architecture, ResNet50, proposed by He et al. [15], is an extremely deep residual network and took first place in the 2015 International Learning and Simulation VRC. The 1.28 million training images were used to train a 50 layer architecture as opposed to a 16 to 19 layer architecture. The size of the input image for this design is $224 \times 224 \times 3$. Following that, numerous CNN architecture variations have been developed in order to offer patients a more accurate diagnosis. On training and testing datasets of breast cancer histopathology images, deep

learning recently demonstrated good classification accuracy [14]. Four fully convolutional networks (FCN) are used in the other study on the detection and classification of breast cancer [11]. The histology scans have also been utilised by CNN to identify colorectal cancer, according to Yoon et al. [50]. Using the complete image as an input, Gao et al. [8] employed deep CNNs to classify interstitial lung diseases (ILD) from CT images.

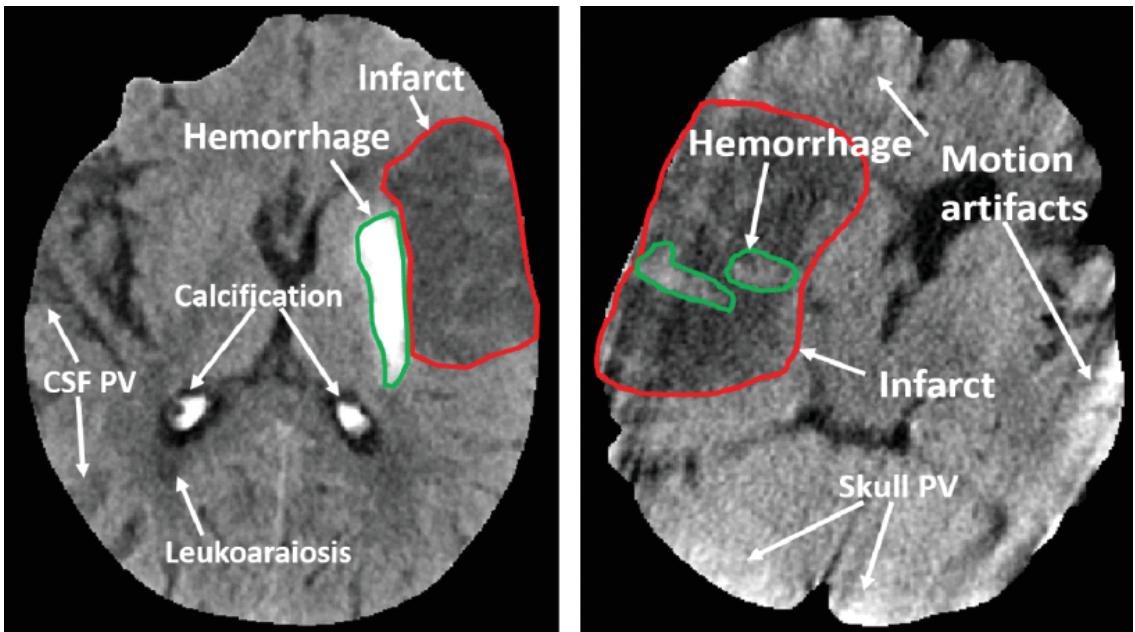


FIGURE 2.1: Samples of CT slices to depict *overlapping infarcts* – infarcts of multiple types at the same image region.

For clinical usefulness, the work on classifying and locating the origin of brain strokes from CT/MR images still needs to be improved. Additionally, the overlapping ischemic and hemorrhagic infarcts that are present on the CT slides as shown in Figure 2.1 create a *multi-label* where each image pixel may belong to more than one class. In order to localise the infarct by volumetric reconstruction and classify brain stroke from CT scan pictures, this research suggests the creation of a new deep learning-based processing pipeline. The infarct will also be able to be localised via the pipeline thanks to volumetric reconstruction. In addition, to the best of the authors' knowledge, this work represents the first effort

to combine volumetric reconstruction and segmentation correction simultaneously in order to enhance demarcation performance and develop a flexible computational model that can be used to derive quantitative diagnostic measures and, with further extensions, for simulating surgical procedures as well.

2.2 PATH PLANNING FOR ROBOTIC MANIPULATORS

Path planning is an essential component of surgical robotics, where precision and accuracy are critical for successful surgical outcomes. In recent years, several studies have proposed various path planning algorithms to improve the accuracy and precision of surgical robots. Ju et al. [21] proposed a hybrid path planning algorithm that combines genetic algorithms and path smoothing techniques to generate a smooth path with fewer sharp turns and corners, reducing the risk of tissue damage during surgical procedures. [31] proposed a generic approach for optimizing the layout of robotic systems to improve the efficiency of surgical procedures. For thrombectomy path planning, Jia Y. et al. [20] proposes a learning-based obstacle avoidance method using Q-learning for path planning in six-axis robotic arms using reinforcement learning for autonomous obstacle avoidance. The strategy prioritizes planning the obstacle avoidance path for the terminal point of the mechanical arm and then uses the calculated terminal path to plan the poses of the mechanical arm. If the mechanical arm cannot avoid obstacles within the limit of the safe distance for points on the terminal path, this strategy records those points as new obstacles and plans a new obstacle avoidance path for the terminal of the mechanical arm. This process is looped until the

correct path is calculated. This method was adopted as a baseline with noteworthy modifications in our work.

Efficient path planning algorithms are essential for real-time robotic applications, where the robot needs to respond quickly to changing environmental conditions. Das et al. [25] proposed a fast path planning algorithm that uses a hierarchical search strategy to generate optimal paths quickly in a two dimensional environment. Zhu et al.

Path planning in environments with obstacles is a challenging problem in robotics, where the robot needs to navigate through a cluttered environment while avoiding obstacles. Bahl et al. [1] proposed a robust path planning algorithm that uses a probabilistic roadmap approach to generate collision-free paths for mobile robots in environments with obstacles. The proposed algorithm can handle large-scale environments with numerous obstacles while ensuring the safety of the robot. Ryu [38] proposed a hierarchical path-planning algorithm for mobile robots using a skeletonization-informed rapidly exploring random tree.

2.3 GAPS IDENTIFIED

Based on the literature survey, there are several gaps in the existing research on NCCT image analysis and path planning for surgical robotics. A wide variety of existing segmentation approaches based on 2D models generally miss out on small regions of hemorrhage. This is likely due to lack of context from neighboring scan slices, which is otherwise available to 3D processing models. Some of the

existing 2D methods, at times, also produce segmentation regions that are starkly diminished or expanded in size.

The purely 3D segmentation models tend to exhibit a smoothing effect of the segmented region. Such smoothing effect is often characteristic of naturally occurring hemorrhagic regions, as opposed to sharp-edged or ridged regions. However, this often causes closely-spaced hemorrhagic regions to get clubbed into a single region when processed by 3D segmentation models.

Although there have been numerous studies proposing various path planning algorithms for surgical robots, there is still a lack of standardization and benchmarking of these algorithms. This makes it challenging to compare the performance of different algorithms and select the most suitable algorithm for a specific surgical procedure. Therefore, there is a need for a standard set of performance metrics and benchmarks that can be used to evaluate and compare the performance of path planning algorithms for surgical robotics.

Moreover, most of the existing path planning algorithms for surgical robotics are designed for offline planning, where the robot generates a path before the surgical procedure begins. However, in some cases, it may be necessary to adjust the path before the surgical procedure due to unforeseen circumstances. Therefore, there is a need for offline path planning algorithms that can adapt to identified changes in the environment prior to execution.

While there have been several studies proposing path planning algorithms for environments with obstacles, most of these algorithms are designed for 2D environments. In surgical robotics, however, the environment is typically 3D, and the obstacles are complex and irregularly shaped. Therefore, there is a need for

path planning algorithms that can handle 3D environments with complex and irregular obstacles.

In addition, while there have been several studies proposing learning-based path planning algorithms for surgical robotics, most of these algorithms are designed for specific surgical procedures and may not be generalizable to other procedures. Therefore, there is a need for generalizable learning-based path planning algorithms that can be applied to a wide range of surgical procedures.

Addressing these gaps in the existing research can lead to the development of more efficient and effective path planning algorithms for surgical robotics, enabling robots to perform surgical procedures with greater precision and accuracy while minimizing the risk of tissue damage and other complications.

2.4 RESEARCH OBJECTIVES

This work is aimed at digitally analyzing NCCT scans of the brain to localize and segment out regions of stroke. This work also develops a Q-learning based path planning algorithm for a 6 degree-of-freedom (DOF) robot operating in an environment with obstacles. The algorithm should be able to generate a collision-free path from a starting point to a target point, while avoiding obstacles in the environment. In implementing this work, the aim was to address the identified research gaps by establishing the following specific objectives:

- The developed segmentation approach should sufficiently resolve closely spaced regions of stroke, potentially combining the effect of

multi-dimensional features to achieve the closest resemblance to ground truth.

- To leverage contextual information to impose geometric constraints when identifying stroke regions from the CT.
- To fuse cross-dimensional futures and potentially overcome the shortcomings of purely 2D and purely 3D segmentation approaches.
- To investigate the use of Q-learning for path planning of a 6-DOF robot in an environment with obstacles.
- To design and implement a path planning algorithm based on q-learning that can handle 6 DOF robots and obstacles of varying shapes and sizes.
- To evaluate the performance of the proposed algorithm in terms of path planning accuracy, efficiency, and robustness in different environments with varying levels of complexity.
- To compare the performance of the proposed algorithm with other state-of-the-art path planning algorithms and evaluate its advantages and limitations.

CHAPTER 3

PROPOSED METHODOLOGY

In recent years, research in deep learning has demonstrated successful application of deep learning and convolutional neural networks for image classification, including medical image analysis. This study proposes the development of a deep neural network-based processing pipeline for classifying different stroke patterns on NCCT images and localizing the infarct for downstream treatment formulation.

Toward solving the problem, NCCT imaging slides of study patients will be collected. Following this, the digital images will be randomly partitioned into two sets, one for model training and another as a held-out test set for the final evaluation of the model. Overlapping slides from a given patient across the training and test sets will be avoided. Each slice will then be manually assessed and its regions will be labeled as either *normal* (0) or *hemorrhagic stroke region* (1) by a trained cardiovascular radiologist. Thus, pixel-level labeled slice images will form the input for training the deep learning -based computational pipeline.

An overview of the proposed solution architecture is presented in Figure 3.1. The computational model will be developed to take in a stack of CT slices as inputs, and to output a prediction probability for each pixel in the input as either belonging to stroke infarct region or not i.e. classes: 1 or 0, respectively. Hence, the hemorrhagic regions in the form of a mask image marks the output of module one of the proposed workflow. As a next step, the authors will facilitate the visualization the model’s stroke region predictions using a web-based hybrid

mobile-workstation tool. This will aid clinicians and radiologists in quickly browsing NCCT scans of the brain, identifying regions of stroke from the scans slice(s) from a given patient, and making quick treatment decisions. The radiologists will then intervene with the help of the visualization tool to identify if surgical intervention is required for treatment, and if so, will identify the target coordinates, and any particular obstacles, to do so. The radiologist input will carried to module two, where a standard pre-defined surgical simulation environment will be augmented with the target and obstacle coordinates. The proposed path-planning algorithm will be used to plan the path of a 6-DOF robotic arm to navigate from its source location to the input target location in the environment for surgical intervention.

Within module one, model parameters including network weights, test for multiple network layers, learning rates, and loss functions will be considered to fine-tune the model and devise a model with optimal parameter settings. An unseen held-out subset of the slice data will be used for independently validating and evaluating the performance of the trained model. This will elucidate the models' ability to recognize the individual segmentation classes. Module two's performance is validated qualitatively in terms of it's ability to reach the target location without striking the obstacles in the environment, and quantitatively in terms of the number of steps it takes to complete the action and the number of episodes it takes to converge and find a path.

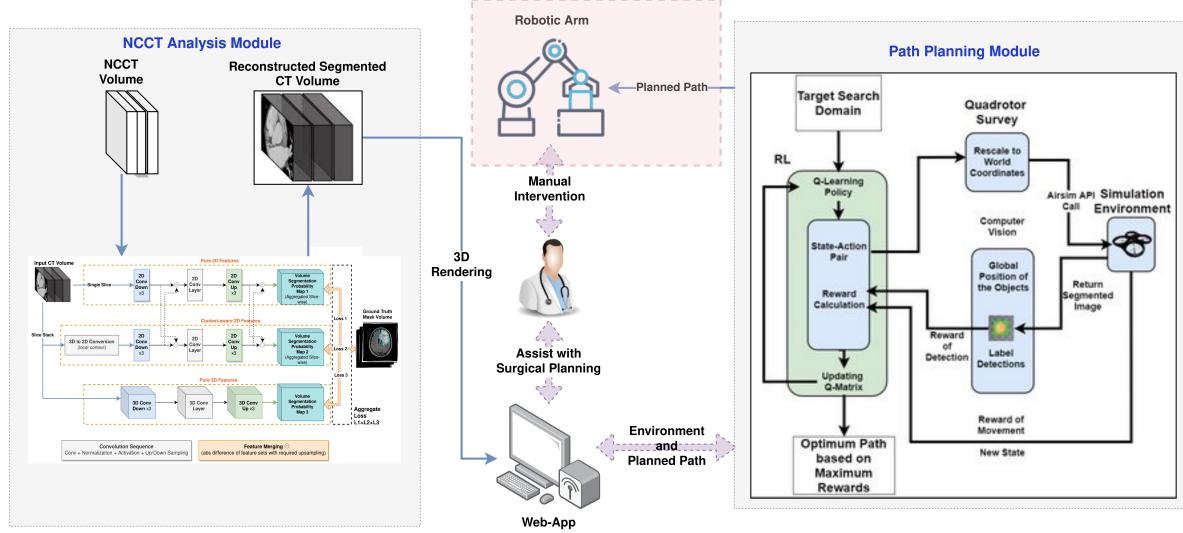


FIGURE 3.1: Architecture of the proposed solution.

3.1 DETECTION AND LOCALIZATION OF HEMORRHAGIC STROKE REGIONS

An overview of the detection and localization module is presented in Figure 3.2. In summary, the following key steps are performed: (1) The input volume stack is presented to the the preprocessing stage; (2) The volume is preprocessed in four steps — skull stripping, tilt correction, smoothing, and non-linear mirroring — and passed to the proposed stroke localization network; (3) The network outputs a pixel-wise probability map to mark stroke and non-stroke regions in the scan; (4) The stroke regions are inferred by bifurcating pixels in the probability map into two classes depending on a threshold probability value; (5) Finally, if the used dataset contains translational parameters that map distance and voxel dimensions on the scan to real-world measurements, quantitative diagnostic measurements such as maximum axis length of the stroke region, volume of the region, etc. will be computed and output.

3.1.1 Data Collection

To develop and evaluate the computational model for detecting distinctive infarct patterns in CTs, volumetric NCCT slices will be acquired in a digital image format, typically DICOM or NIfTI. Following this, the images will be randomly partitioned into two sets: (a) one for *model building* – i.e., training and validation (about three-fourths of the dataset), and (b) for *model evaluation* (about one-fourth of the dataset).

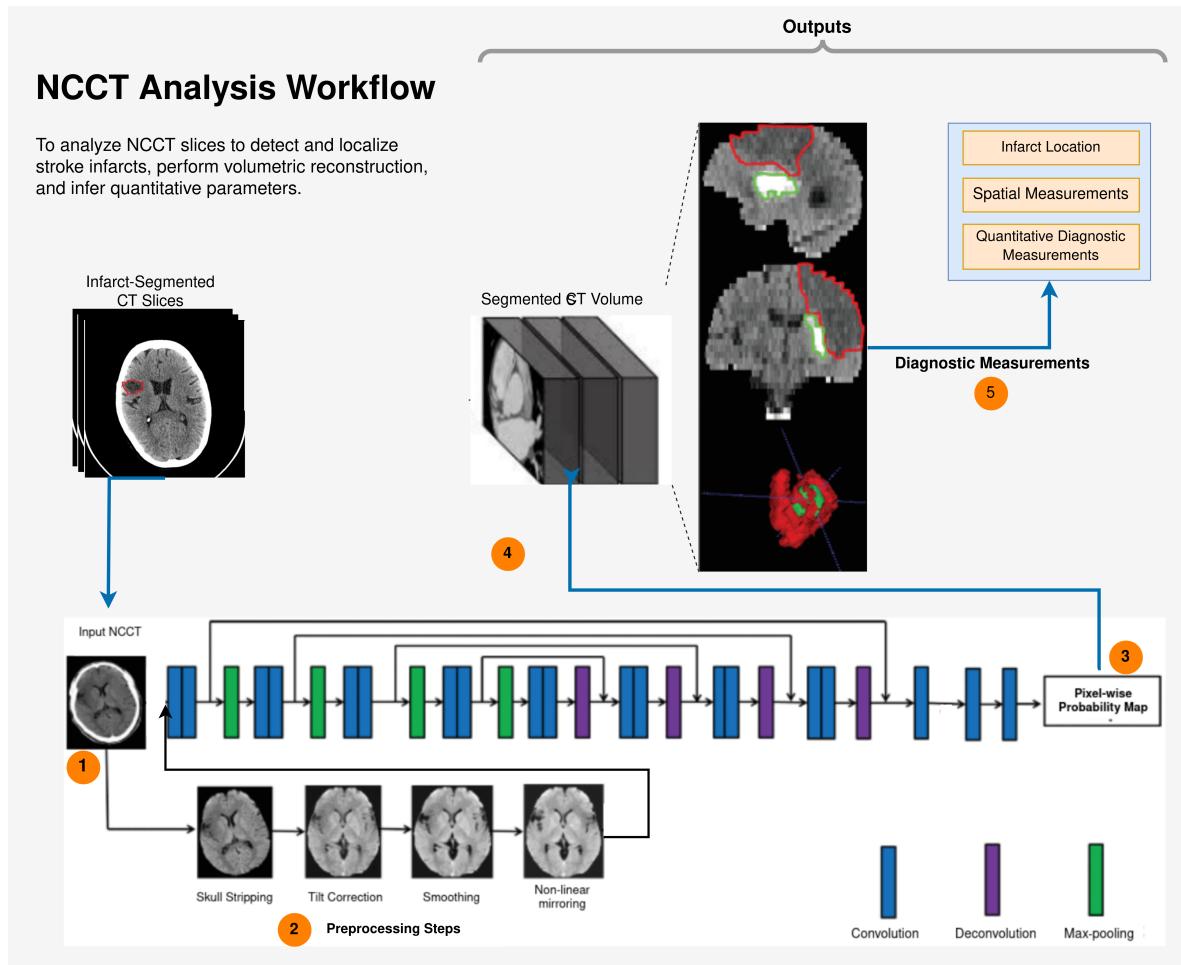


FIGURE 3.2: Overview of the NCCT analysis workflows.

3.1.2 Slice-level Annotation to Establish Ground Truth

All NCCT slice images will be manually labeled by one or more medical experts. Each image corresponding to the model building dataset will be annotated with scan regions as either one of the two patterns or normal/non-applicable. These are: *normal (0)* and *stroke region (1)*. Ideally, multiple independent annotation results by experts would be utilized and inter-rater biases corrected to ensure high quality of the ground truth. For the purpose of annotations, a granulated user-friendly tool will be provided to the medical expert(s) to perform manual annotation at the super-pixel image region level, with the ability to select large regions of the image at a time and label them. Furthermore, an unsupervised clustering-based segmentation algorithm will be applied to facilitate easy manual annotation of infarct regions on the NCCT slices. This will save time and effort on their part. In addition, case-specific comments will be collected for downstream evaluation and analysis of the developed processing framework. The test set will be used, once the model development is complete, to evaluate its performance and compare it against that of expert annotators.

The unsupervised segmentation algorithm is presented in Figure 3.3. This algorithm takes slices of NCCT scans and performs the following sequence of steps: skull stripping to remove the bone regions from the scan, a normalization pre-processing step by histogram equalization to ensure color uniformity across scans, image feature extraction using a pre-trained ResNet101 architecture, clustering based on the extracted features, grouping the foreground cluster points as stroke region mask, and output the final stroke-segmented scan slice. The proposed method achieves reasonable performance for a preliminary round of segmentation to facilitate correction of segmentation regions by radiologists; it

achieved a dice score coefficient of 32.72% and a Mean IoU score of 21.09% on the same held-out test set used for model training experiments of the proposed segmentation architecture (see chapter 4).

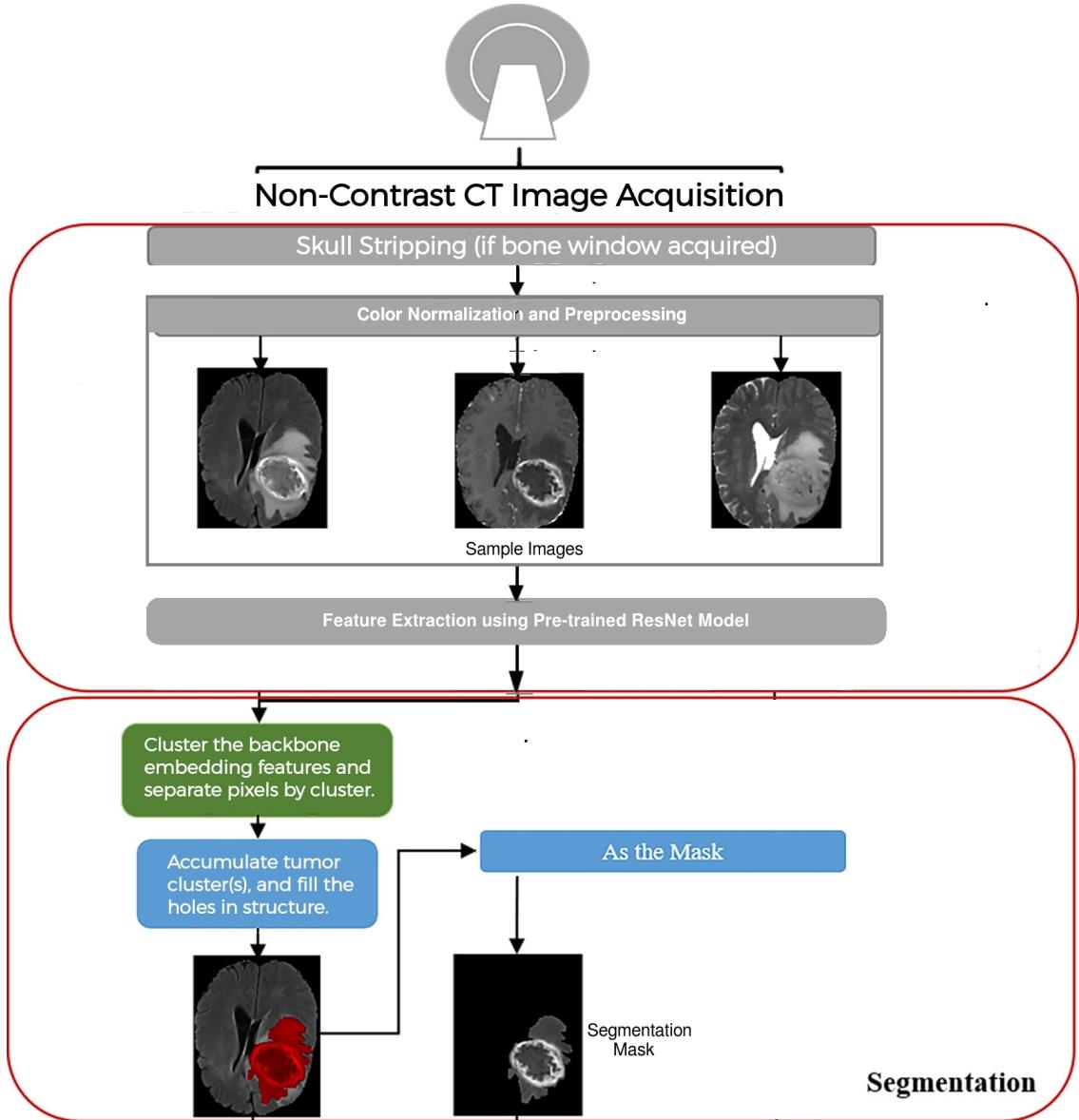


FIGURE 3.3: An unsupervised approach to segment stroke regions from brain CTs.

3.1.3 Data Pre-processing

The input data in the form of CT slices is processed by a computational

processing pipeline that comprises deep learning network to analyze images and draw inferences. Hence initially, basic pre-processing methods to ensure consistency of image quality and exposure statistics across the dataset were applied. The pre-processing methods applied include: automatic resizing of training images to contain a fixed dimension of pixels, skull stripping, tilt correction, smoothing, non-linear mirroring, and color normalization. The steps of the color normalization process for two sample scan slices is shown in Figure 3.4. Augmentation is then performed to randomly flip and rotate them along the horizontal and vertical axes to obtain variability in the training data, and to ensure robust model development, among other benefits.

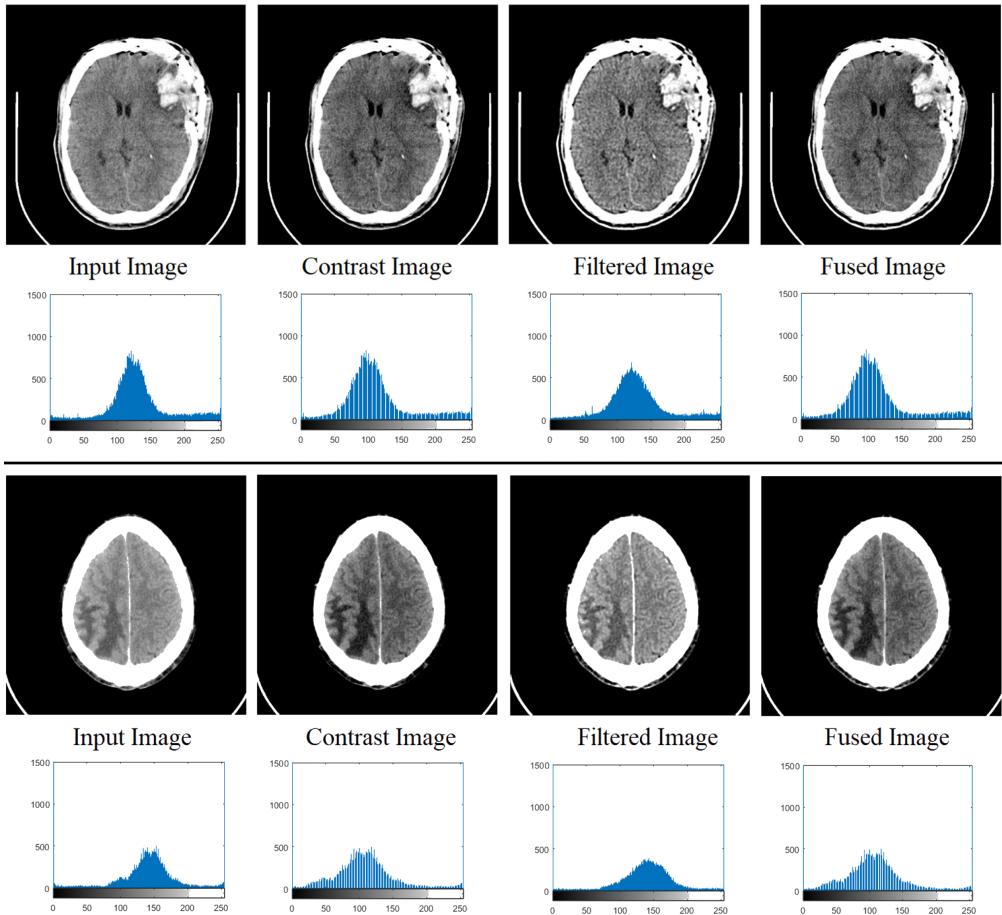


FIGURE 3.4: Pre-processing methods to apply on the NCCT slices before passing them to the neural network and computational analysis pipeline to correct artifacts.

3.1.4 Detection and Segmentation Model Building

Before training the classification and segmentation models, 90% of the model building data were randomly selected for training and 10% for internal validation. The models were implemented using the deep learning library Torch available for Python. Torch is a high-level neural networks API, capable of running on top of various backend architectures. The models were trained under a supervised setting, with NCCT slices fed as input, and outputs corrected against their corresponding expert-annotated and inter-rater corrected NCCT slices. Different tests were conducted to optimize for the number of epochs, initial learning rates as well as the optimal depth of the neural network and other functionally configurable portions of the computational pipeline. The output of the models were then configured to output probability vectors p per stroke type, normalized to 1. Finally, the learning curves (accuracy vs. no. of iterations) for both training and testing were plotted to determine if there is a generalization of the deep learning network model(s) to the unseen validation/test data i.e., no over-fitting has occurred. Two state-of-the-art slice-based image segmentation neural networks, namely Slice-Aware Net (SA-Net) [27] and M-Net [26] were adopted as improved models. In addition, a combination of the two models was also implemented to combine the strengths of these state-of-the-art approaches – long-range dependency capturing ability of SA-Net and spatial context capturing ability of M-Net.

The proposed segmentation network architecture is presented in Figure 3.5. The network is composed of three image processing paths, that each process the scans in a different dimensional context whilst merging features across paths for multidimensional analysis.

Path one extracts each slice of the scan as a 2D image, and performs convolutional downsampling and upsampling using 2D convolutions. This path processes scans as purely two-dimensional data. It outputs a segmentation probability map for the input slice.

Path two extracts a stack of slices as a 3D volume. The stack contains neighboring slices present before and after a target slice in the NCCT volume. It then performs a matrix operation to convert it to a 2D representation. Following this, the 2D representation is processed as though it is a simple two-dimensional input, using 2D convolutions. This path allows local context to be incorporated while processing the target slice. It outputs a segmentation probability map for the target slice.

Path three processes either the entire volume, or a portion thereof, depending on a hyperparameter that controls the size of processing volumes. This is required for resource-constrained machines that may not be able to process the entire volume in one stretch. The path processes volumes using 3D convolutions, and outputs a segmentation probability map for the entire volume.

The outputs of paths one and two are collected slice-wise for the entire CT volume. The output of path three is collected for the entire volume or for each set of slices and stacked together, depending on the hyperparameter setting. This produces three separate volumetric segmentation probability maps for the entire input NCCT scan. Finally, the probability values are averaged across the three maps, and a threshold value is used to compute the output segmentation mask. During training, the probability values are directly used, before thresholding, to compute the loss and backpropagate.

The representations after each level of convolution in each path is merged. This step is called feature merging, and is performed by computing the absolute difference of the corresponding feature representation matrices after rescaling their dimensions as appropriate. This allows information flow between paths and enables multidimensional feature fusion for a more informed segmentation pipeline.

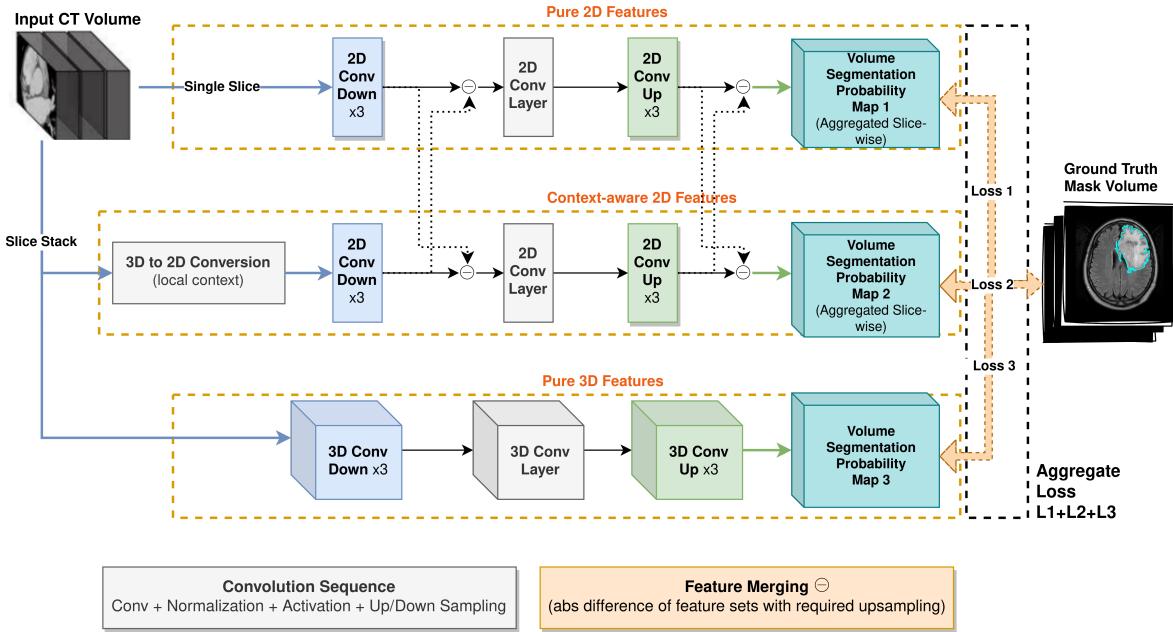


FIGURE 3.5: Proposed network architecture for scan segmentation.

3.1.5 Visualization of Hemorrhagic Regions on NCCT Scans

The processed and region-predicted NCCT slices will be visualized along with the segmented and identified stroke infarct regions using a visualization tool. This visualization will allow clinicians to easily browse scans and make treatment decisions accordingly. Furthermore, it will allow them to intervene to configure parameters for the robotic path-planning module, if surgical intervention is necessary.

3.1.6 Diagnostic Inference from Volumes

The ultimate goal is to localize stroke locations in the spatial context. In addition, it is proposed to estimate quantitative diagnostic measures if the NCCT scans contain associated metadata that can be used to map dimensions on the scan to the real-world. In such cases, the annotated volume of the patient’s brain can be easily exploited to draw out any form of geometric, spatial, or structural measurements, and even a combination thereof. Furthermore, the stoke regions can be individually quantified to present summary inferences of the patient diagnosis to help clinicians devise an ideal treatment strategy.

3.1.7 Validation from Radiologists

At the completion of the complete processing pipeline, two sets of validations were performed to ensure correctness of the overall computational pipeline – (1) *the volume-level annotations* were validated by radiologists, and (2) *the slice-level annotations* of the held-out test-set were compared to the model predictions. *Evaluation 1* is qualitative in nature, owing to the tediousness of manually annotating volumetric regions. However, this evaluation will be supported by *evaluation 2*. In conjunction, the two sets of validations provide sufficient evidence to verify the complete proposed computational pipeline. To evaluate the slice-level performance, the model’s labels were compared against those of the radiologists’ by calculating an inter-rater reliability metric called Cohen’s kappa score [24], as it has been adapted as a standard metric when it comes to annotation of medical images. Thus, between every two sets of annotations from radiologists and/or the computational pipeline, both ρ_d scores

and kappa score per class were computed. K_{pd} refers to the percentage of NCCT slices in which two annotators agreed on the same annotation. Kappa scores per infarct class, on the other hand, estimates the detection of a given infarct pattern or type between two sets of annotations.

3.2 ROBOTIC PATH PLANNING MODULE

Path planning is a crucial task in robotic applications, especially in surgical robotics, where precision and accuracy are essential for successful surgical outcomes. In this project, a q-learning based path planning algorithm is developed for a 6 degree-of-freedom (DOF) robot operating in an environment with obstacles.

To model robotic environments based on quantitative parameters measured from the proposed NCCT analysis pipeline, experiments were conducted to determine the most feasible simulation workflows and perform path-planning for a 6-DOF arm. Based on this review, suitable simulation gyms for this project include:

- *OpenAI Gym* is a comprehensive collection of environments to train and test reinforcement learning algorithms.
- *Panda Gym* [7] provides a set of RL environments integrated with the OpenAI Gym. It provisions five tasks, namely reach, push, slide, pick place, and stack.

- *Robo Gym* [33] is a unified setup for simulation and real environments, allowing seamless transfer from training to application. Particularly known for its distributive capabilities, it provides further scope for extension.
- *RVCTools* [5] is a MATLAB toolbox for robotic manipulators that provides a collection of functions for robot modeling, control, and visualization. It includes tools for kinematics, dynamics, trajectory generation, and motion planning of robotic systems. The toolbox supports various types of robots, including serial and parallel manipulators, and provides functions for forward and inverse kinematics, Jacobians, and inverse dynamics.

The initial attempts at path planning for the 6-DOF robot involved deep reinforcement learning (DeepRL) techniques. However, the training process was unstable and the resulting policy was not reliable. Therefore, an alternative approach based on Q-learning was adopted. The Q-learning algorithm was modified to include obstacle avoidance as part of the learning objective. Additionally, an annealing strategy was proposed to encourage the exploration of new paths during the training process. Furthermore, new distance metrics were introduced to improve the convergence of the algorithm and to generate more optimal paths.

3.2.1 Deep Reinforcement Learning

Deep reinforcement learning (DeepRL) is a subset of machine learning that involves training an artificial agent to interact with an environment in order to learn a policy that maximizes a reward signal. The agent is typically represented

as a neural network, and the learning process involves adjusting the weights of the network to maximize the expected reward over a large number of interactions with the environment. In robotic applications, DeepRL has been used to learn policies for a variety of tasks, including grasping, reaching, and locomotion, among others. The main advantage of DeepRL is that it can learn policies from raw sensor data, without the need for hand-engineered features or models of the environment. However, the success of DeepRL in practice depends heavily on the choice of reward function, the architecture of the neural network, and the choice of hyperparameters.

To evaluate the feasibility of DeepRL, the Panda Reach environment was selected, which involves moving the end-effector of the robotic arm to a specific target location. This environment was chosen due to its relative simplicity and relevance to the path planning problem. The environment has a continuous action space and a continuous observation space, which makes it suitable for deep reinforcement learning techniques. The depicted environment in Figure 3.6 shows the target represented by the red circle.

3.2.1.1 Experiments

Two state-of-the-art deep reinforcement learning algorithms, DDPG and TD3, were employed to solve the Panda Reach environment.

- *Deep Deterministic Policy Gradient (DDPG)* is an off-policy algorithm for continuous action spaces, which extends the Q-learning algorithm to actor-critic methods. It consists of an actor network that predicts the action to take given the current state, and a critic network that estimates the Q-value

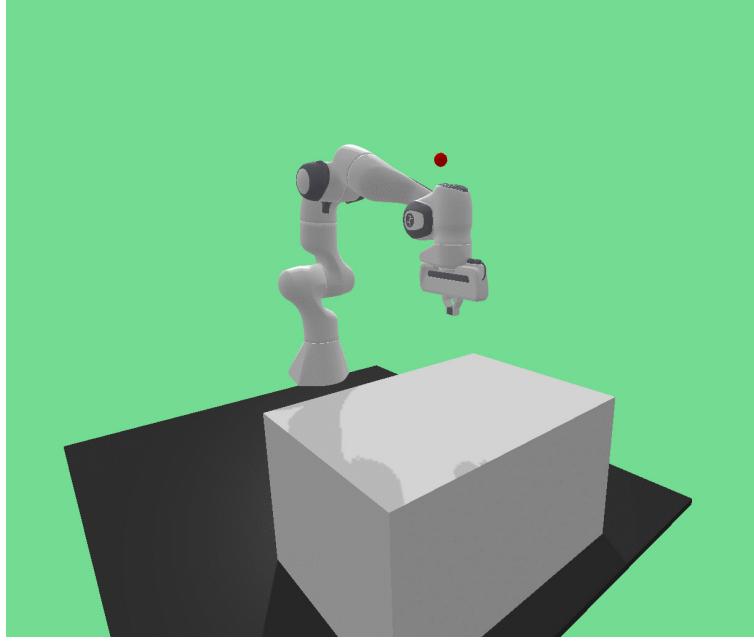


FIGURE 3.6: The PandaReach-v2 robotic environment.

of the action-state pair. The actor is trained to maximize the estimated Q-value, and the critic is trained to minimize the mean-squared error between its estimated Q-value and the true Q-value.

- *Twin Delayed DDPG (TD3)* is an improved version of DDPG, which introduces three key modifications: clipped double Q-learning, delayed policy updates, and target policy smoothing. Clipped double Q-learning mitigates the overestimation bias inherent in traditional Q-learning, delayed policy updates prevent the actor from chasing moving targets, and target policy smoothing regularizes the output of the actor, making it more robust to small perturbations.

The experimental parameters for both experiments are shown in Table 3.1 In both experiments, the models were trained on an NVIDIA Tesla V100 GPU with a total training time of approximately 15 hours for each algorithm. The performance of

the trained models was evaluated using the mean success rate over 100 evaluation episodes.

Algorithm	Episodes	Max Steps / Episode	γ	Model Description
DDPG	1,000,000	2000	0.95	2×256 hidden layers
TD3	1,000,000	2000	0.95	3×512 hidden layers

TABLE 3.1: Experimental parameters for DDPG and TD3.

3.2.1.2 Pitfalls

The experimental results revealed that both the DDPG and TD3 algorithms performed poorly on the task of controlling the robotic arm, as shown in Table 3.2. The mean reward over 100 evaluation episodes was negative, and the success rate was very low, indicating that the robotic arm was unable to accomplish the desired task with the trained models. Moreover, the arm’s movement was observed to be shaky during the experiments. These results indicate that deep reinforcement learning may not be suitable for precise robotic applications such as surgery.

Experiment	Mean Reward	Success Rate (100 episodes)
DDPG	-2.3	0
TD3	-2.7	2

TABLE 3.2: Results for DDPG and TD3 experiments.

3.2.2 Q-Learning

Q-learning is a model-free, off-policy reinforcement learning algorithm that learns the value of an action in a particular state. The algorithm attempts to learn an optimal policy that maximizes the cumulative reward over a sequence of actions.

The goal of Q-learning is to learn a function $Q(s, a)$, where s is the current state and a is the action taken in that state, that maps the expected reward of taking action a in state s . The optimal policy can be obtained by choosing the action with the highest expected reward in each state.

The Q-learning algorithm iteratively updates the estimate of $Q(s, a)$ by applying the following update rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

where s_t is the current state, a_t is the action taken in that state, r_{t+1} is the reward received after taking that action and transitioning to the next state s_{t+1} , α is the learning rate that controls the step size of the update, and γ is the discount factor that controls the importance of future rewards.

Building atop the proposed Q-learning strategy in [20], and deriving from methods for obstacle avoidance in robotics, such as the artificial potential field-based industrial robot station conversion obstacle avoidance system, genetic algorithms, and RBF neural networks have limitations in complex environments and high-dimensional multi-constraint path planning problems. The approach prioritizes planning the obstacle avoidance path for the terminal point of the mechanical arm and then uses the calculated terminal path to plan the poses of the mechanical arm. If the mechanical arm cannot avoid obstacles within the limit of the safe distance for points on the terminal path, the strategy records those points as new obstacles and plans a new obstacle avoidance path for the terminal of the mechanical arm, looping this process until the correct path is calculated.

An overview of the path-planning workflow is presented in Figure 3.7. The proposed method makes use of a novel distance function and simulated annealing techniques for exploratory actions.

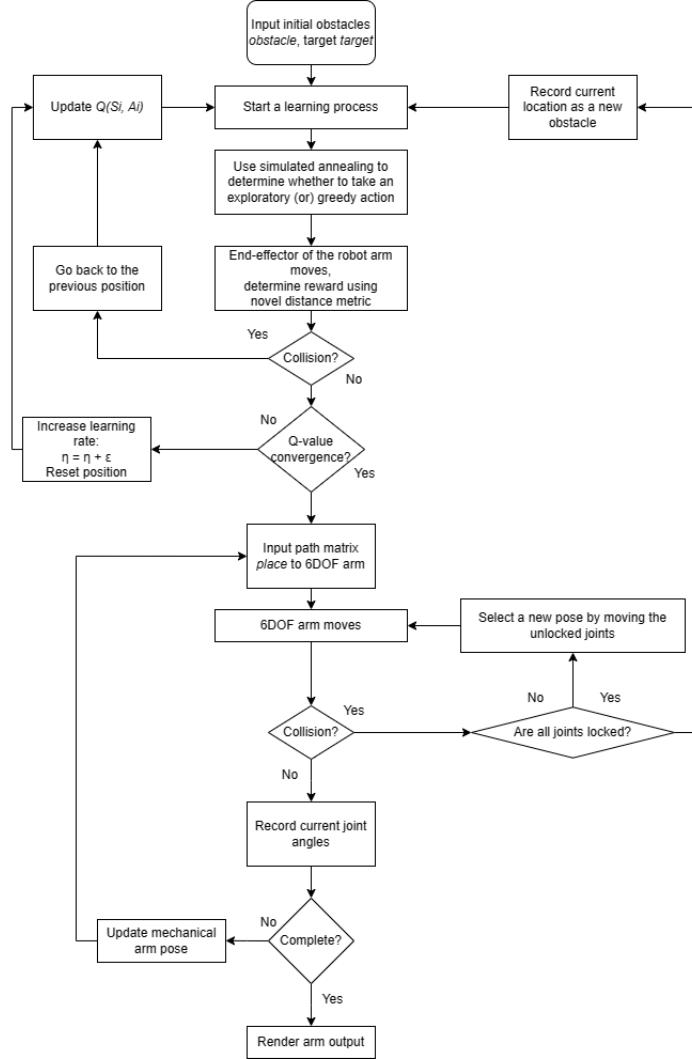


FIGURE 3.7: Path planning workflow based on Q-learning for the 6DOF robotic arm.

Figure 3.7 depicts learning process with simulated annealing to control the movements of a 6-degree-of-freedom (6DOF) robot arm in order to reach a target while avoiding obstacles and consists of two phases. In the first phase, The end-effector of the robot arm moves and a reward is determined using a novel distance metric. If a collision occurs, the algorithm goes back to the previous position and updates the $Q(S_i, A_i)$ value. If Q-value convergence is not achieved

at the end of a fixed number of steps, the learning rate is increased, and the process is restarted. In the second phase, the algorithm inputs a path matrix for the 6DOF arm and checks for collisions. If the arm collides with an obstacle and all positions are locked, the position of the end-effector is recorded as an obstacle and the first phase is restarted. Otherwise, a new pose is selected by moving the unlocked joints. On the other hand, if there is no collision, the algorithm updates the mechanical arm pose until the process is complete. Finally, the arm output is rendered.

3.2.3 Enhancements to Q-Learning

The two modifications to the standard Q-learning approach for the robotic arm control task include the use of annealing for exploratory actions and the incorporation of a distance metric to penalize the agent for hitting obstacles. The annealing process gradually decreases the exploration rate, allowing the agent to focus on exploiting its current knowledge. The distance metric uses a custom algorithm that calculates the maximum distance between the current and previous arm positions and the target point, while also incorporating a penalty for collisions with obstacles. This modified approach is expected to improve the performance of the Q-learning algorithm for the robotic arm control task.

3.2.3.1 Simulated Annealing

A temperature-based simulated annealing strategy is used to decide between greedy (maximizing reward) and exploratory (random) actions.

Let $E(s)$ be the energy (or cost) associated with state s , ΔE be the energy difference between the current state and the proposed next state, and $T(t)$ be the temperature at iteration t . The probability of accepting a worse state at temperature $T(t)$ is given by the Boltzmann distribution:

$$P(\Delta E, T(t)) = \exp\left(-\frac{\Delta E}{T(t)}\right) \quad (3.1)$$

To decide whether to take a greedy or exploratory action, start with a high initial temperature T_0 and gradually decrease it over time according to a cooling schedule, such as:

$$T(t) = T_0 \cdot \alpha^t \quad (3.2)$$

where α is the cooling rate, t is the iteration number, and $T(t)$ decreases exponentially as t increases. At each iteration, a random neighboring state of the current state can be generated, and the energy difference ΔE can be computed. If $\Delta E \leq 0$, the new state is accepted as the current state. Otherwise, the new state is accepted with probability $P(\Delta E, T(t))$.

3.2.3.2 Distance Metric

The distance is calculated using a custom metric that incorporates the distance between the current and previous arm positions and the target point. The function uses vector algebra to determine the shortest distance between the line segment and the third point.

Given two vectors \mathbf{u} and \mathbf{v} with n elements, the maximum distance between them using the Chebyshev and Minkowski distance metrics, with p defined as the difference between the Euclidean distances of the current arm position and previous arm position to the target point, described in Equation 3.3.

$$\max \left(\max_{i=1}^n |u_i - v_i|, \left(\sum_{i=1}^n |u_i - v_i|^p \right)^{1/p} - \left(\sum_{i=1}^n |u'_i - v'_i|^p \right)^{1/p} \right) \quad (3.3)$$

where \mathbf{u}' and \mathbf{v}' are the previous arm position and target, respectively, $\|\cdot\|_2$ denotes the Euclidean norm. Further, p is computed using Equation 3.4.

$$\|\text{current arm position} - \text{target}\|^2 - \|\text{previous arm position} - \text{target}\|^2 \quad (3.4)$$

The distance metric used in this study takes into account the distance between the current and previous arm positions and the target point. The intuition behind this approach is to penalize the agent for taking very small steps towards the target and to discourage it from hitting obstacles. To achieve this, a parameter, denoted as p , is introduced, which encodes the obstacle in some sense. If the arm hits an obstacle and comes back, it is penalized even more. Additionally, the Chebyshev distance metric is used to prevent the agent from taking only very small steps. This metric calculates the maximum absolute difference between any two corresponding elements of the current and previous arm positions and the target point. Together, these considerations provide a more effective distance metric for controlling the robotic arm in the presence of obstacles.

3.3 PROOFS OF HYPOTHESES USED

Portions of the adopted methodology in the proposed solution was designed based on three hypotheses. First, a slice-wise segmentation approach was used amidst several other approaches in literature owing to its advantages in model inference time in the context of deep learning models and spatial context awareness, among others. Second, to facilitate easy manual annotation of NCCT volumes by experts, an approximate unsupervised segmentation algorithm was designed. A primitive proof of the feasibility and applicability of these methods is presented in this section. A novel distance metric has been employed for the purpose of path planning of a robotic arm in environments containing obstacles. Our proof shows that this distance metric exhibits faster convergence rates as compared to the commonly used Euclidean distance metric.

3.3.1 On The Effectiveness Of Slice-wise Segmentation

3.3.1.1 Statement

Slice-wise segmentation is at least as effective as global volumetric segmentation of the CT slices to identify infarct regions.

3.3.1.2 Explanation

One begins by noting that many structures in medical imaging are naturally represented as a stack of 2D images, with the relevant features of the structure spread out across multiple slices. For example, the edges of the lungs are often

visible on multiple slices, and may be difficult to identify on any individual slice. Examining the entire stack of slices enables easier and more accurate identification of edges. Therefore, slice-by-slice segmentation can be an effective method for segmenting these structures.

To make this more concrete, let's consider a simplified example of a 3D CT image of a cube. Assuming that the objective is to segment the cube from the background, one could apply a thresholding algorithm to the entire 3D volume at once, but this may not be effective if the cube is not easily distinguishable from the background. Instead, one could use slice-by-slice segmentation. One begins by segmenting the first slice using a thresholding algorithm. Then, move on to the second slice, and use the segmentation of the first slice as a prior to guide the segmentation of the second slice. Specifically, one can use the segmentation of the first slice to identify a range of intensity values that are likely to correspond to the cube, and then apply a threshold within this range to the second slice. One repeats this process for each subsequent slice, using the previous segmentations as priors to guide the segmentation of the current slice.

The key advantage of this approach is that it can be more robust to noise and other artifacts that may be present in the image. If one were to apply a global thresholding algorithm to the entire 3D volume at once, one may include or exclude voxels that should be part of the cube due to variations in the intensity values across the volume. However, by segmenting each slice individually and using the previous segmentations as priors, one can incorporate more information about the structure into the segmentation process, and potentially achieve better results.

Therefore, one can conclude that slice-by-slice segmentation can be an effective way to segment 3D CT images, particularly for structures that are naturally represented as a stack of 2D images. By segmenting each slice individually and using the previous segmentations as priors, one can incorporate more information about the structure into the segmentation process and potentially achieve better results than by applying a global thresholding algorithm to the entire 3D volume at once.

3.3.1.3 Proof 1 – By Induction

Assumptions

1. The CT scan is of a single anatomical structure, such as a lung, liver, brain, or kidney.
2. The structure is contiguous, meaning that there are no gaps or holes in the 3D volume.
3. The structure has a consistent appearance across all slices, meaning that it looks the same in each 2D image.

Definitions

1. A voxel is a 3D pixel in the CT image, with a certain intensity value.
2. A segmentation is a labeling of voxels as belonging to either the structure of interest or the background.

Let $I(x,y,z)$ be a 3D CT image, where x,y,z denote the spatial coordinates. Our goal is to segment a target structure S from the background. Assume that S is connected and has a reasonably smooth boundary in the image.

Suppose one applies a global thresholding algorithm to the entire 3D volume at once, using a threshold value T . Let S_T be the resulting segmentation, defined as:

$$S_T(x,y,z) = \begin{cases} 1 & \text{if } I(x,y,z) > T \\ 0 & \text{otherwise} \end{cases}$$

Note that S_T is a binary image that indicates which voxels belong to the target structure.

Now, let us consider slice-by-slice segmentation. Begin by segmenting the first slice, which is assumed to be parallel to the xy plane. Let $S_1(x,y)$ be the resulting segmentation of the first slice, defined as:

$$S_1(x,y) = \begin{cases} 1 & \text{if } I(x,y,1) > T \\ 0 & \text{otherwise} \end{cases}$$

Note that S_1 is a binary image that indicates which pixels in the first slice belong to the target structure.

Subsequently, the second slice is processed, and the segmentation of the first slice is utilized as a prior to guide the segmentation of the second slice. Specifically, define a prior probability map $P(x,y)$, where $P(x,y)$ is proportional to the

probability that a voxel with spatial coordinates $(x, y, 1)$ belongs to the target structure. Define $P(x, y)$ as:

$$P(x, y) = \frac{1}{Z_1} \sum_{i,j} S_1(i, j) \exp\left(-\alpha(x-i)^2 - \beta(y-j)^2\right)$$

where Z is a normalization constant, α and β are parameters that control the spatial smoothness of the prior, and the sum is taken over all pixels (i, j) in the first slice. Note that the prior probability map is essentially a smoothed version of the segmentation of the first slice, with the smoothing controlled by the parameters α and β .

One can then use the prior probability map to guide the segmentation of the second slice. Specifically, define the segmentation of the second slice, $S_2(x, y)$, as:

$$S_2(x, y) = \begin{cases} 1 & \text{if } I(x, y, 2) > TP(x, y) \\ 0 & \text{otherwise} \end{cases}$$

It should be noted that the prior probability map $P(x, y)$ has been utilized to define a local threshold value $TP(x, y)$ for each voxel in the second slice.

One can repeat this process for each subsequent slice, using the previous segmentations as priors to guide the segmentation of the current slice. Specifically, for the k th slice, define the prior probability map $P(x, y)$ as:

$$P(x, y) = \frac{1}{Z_{k-1}} \sum_{i,j} S_{k-1}(i, j) \exp\left(-\alpha(x-i)^2 - \beta(y-j)^2\right)$$

and the segmentation of the k th slice, $S_k(x, y)$, as:

$$S_k(x, y) = \begin{cases} 1 & \text{if } I(x, y, k) > TP(x, y) \\ 0 & \text{otherwise} \end{cases}$$

The final segmentation of the target structure, $S_{final}(x, y, z)$, is the union of all slice segmentations:

$$S_{final}(x, y, z) = \bigcup_{k=1}^n S_k(x, y) \delta_{z, k}$$

where $\delta_{z, k}$ is the Kronecker delta function that equals 1 if $z = k$ and 0 otherwise.

The claim is that this slice-by-slice segmentation algorithm is equivalent to the global thresholding algorithm. To demonstrate this, it is necessary to show that $S_{final}(x, y, z)$ is equivalent to $S_T(x, y, z)$ for all voxels (x, y, z) .

First, note that by construction, $S_1(x, y)$ is a subset of $S_T(x, y, 1)$ for all (x, y) . That is, the pixels that are classified as belonging to the target structure in the first slice using the slice-by-slice algorithm are a subset of the pixels that are classified as belonging to the target structure using the global thresholding algorithm.

Now, suppose that for some $k > 1$, $S_{k-1}(x, y)$ is a subset of $S_T(x, y, k-1)$ for all (x, y) . It is claimed that this implies that $S_k(x, y)$ is a subset of $S_T(x, y, k)$ for all (x, y) .

To see why, consider a voxel (x, y, k) that belongs to the target structure. By definition of S_T , $I(x, y, k) > T$. Also, by assumption, $S_{k-1}(i, j) = 1$ for all (i, j)

such that $(i, j, k - 1)$ belongs to the target structure. Therefore, the prior probability map $P(x, y)$ is nonzero in a neighborhood of (x, y) , and the local threshold value $TP(x, y)$ is greater than T . Hence, $S_k(x, y) = 1$ for this voxel.

Conversely, suppose that for some (x, y, k) , $S_k(x, y) = 1$ but $S_T(x, y, k) = 0$. Then, by definition of S_T , $I(x, y, k) \leq T$. But this means that the local threshold value $TP(x, y)$ is also less than or equal to T , since $P(x, y) \leq 1$. Therefore, it must be the case that $S_k(x, y) = 0$, which contradicts our assumption.

Therefore, for all k , $S_k(x, y)$ is a subset of $S_T(x, y, k)$ for all (x, y) . Hence, the final segmentation $S_{final}(x, y, z)$ is a subset of $S_T(x, y, z)$ for all voxels (x, y, z) . By a similar argument, one can show that $S_T(x, y, z)$ is a subset of $S_{final}(x, y, z)$, so the two segmentations are equal.

This completes the proof that the slice-by-slice segmentation algorithm is equivalent to the global thresholding algorithm.

3.3.1.4 Proof 2 – By Contradiction

This theorem will be proved by contradiction. Suppose that slice-by-slice segmentation does not work, i.e., that it produces an invalid segmentation of the entire structure. Then, there must be at least one voxel in the structure that is mislabeled by the segmentation. Let v be such a voxel. Consider the slice that contains v . Since the structure is contiguous, there must be some neighboring voxels in the same slice that are also part of the structure. Let w be one of these voxels.

Case 1. The voxel v is mislabeled as background, but it should be part of the structure. In this case, the intensity value of v must be similar to the intensity value of w , since they are both part of the same structure. Therefore, if one were to use a global thresholding method to segment the entire 3D volume at once, both v and w would be labeled as part of the structure. Nonetheless, since slice-by-slice segmentation is being performed, voxel v may have been mislabeled due to the threshold utilized on that specific slice being inappropriate. Therefore, this case contradicts our assumption that the structure has a consistent appearance across all slices.

Case 2. The voxel v is mislabeled as part of the structure, but it should be background. In this case, the intensity value of v must be different from the intensity value of w , since w is part of the structure and v is not. Therefore, if one were to use a global thresholding method to segment the entire 3D volume at once, v would be correctly labeled as background. Nonetheless, since slice-by-slice segmentation is being performed, voxel v may have been mislabeled due to the threshold utilized on that specific slice being inappropriate. Therefore, this case contradicts our assumption that the structure is contiguous.

Since both cases lead to a contradiction, it has been shown that slice-by-slice segmentation of a contiguous anatomical structure in a 3D CT scan works, given the assumptions above.

3.3.2 On The Convergence Of A Novel Distance Metric For Path Planning In Environments With Obstacles

3.3.2.1 Statement

The new distance metric $D(x(t), y, p)$ defined as $\max(D_C(x(t), y), D_M(x(t), y, p_2))$ converges faster than $D_E(x(t), y)$ in an environment with obstacles.

3.3.2.2 Explanation

The Euclidean distance metric, which represents the shortest distance between two points in a straight line, is used to measure the distance between the arm and the target position. However, in environments where obstacles are present, this metric may not be suitable, as it does not account for the presence of obstacles.

To address this limitation, a new metric that combines the Euclidean distance with two other distance metrics, namely the Chebyshev distance and the Minkowski distance. The Chebyshev distance is the maximum difference between the corresponding components of two vectors, and the Minkowski distance is a generalized form of the Euclidean distance that takes a parameter.

The proposed metric takes into account the current and previous Euclidean distances to the target and is obtained by taking the maximum of the Chebyshev distance and the Minkowski distance with a parameter. Our analysis shows that this new metric exhibits faster convergence than the Euclidean distance in environments with obstacles.

To compare the convergence rates between the new metric and the Euclidean distance, their time derivatives are computed. Analysis of the results indicates that the rate of change of the new metric is greater than or equal to that of the Euclidean distance.

3.3.2.3 Proof

Let $x(t)$ be the position of the autonomous arm at time t , and let y be the target position. Define the Euclidean distance between $x(t)$ and y as:

$$D_E(x(t), y) = \sqrt{\sum_{i=1}^n (y_i - x_i(t))^2}$$

Define the Chebyshev distance as:

$$D_C(x(t), y) = \max_{i=1}^n |x_i(t) - y_i|$$

And the Minkowski distance with parameter p_1 as:

$$D_M(x(t), y, p_1) = \left(\sum_{i=1}^n |x_i(t) - y_i|^{p_1} \right)^{\frac{1}{p_1}}$$

Now, define a new distance metric:

$$D(x(t), y, p) = \max\{D_C(x(t), y), D_M(x(t), y, p_2)\}$$

where $p = \max(p_1, p_2)$ and p_2 is the parameter for the Minkowski distance that depends on the current and previous Euclidean distances to the target, as defined in the problem statement.

The objective is to demonstrate that $D(x(t), y, p)$ converges faster than $D_E(x(t), y)$ in an obstacle-filled environment.

To do this, first note that $D(x(t), y, p)$ is always greater than or equal to $D_E(x(t), y)$, since both $D_C(x(t), y)$ and $D_M(x(t), y, p_2)$ are non-negative. Therefore, it is only needed to show that $D(x(t), y, p)$ converges faster than $D_E(x(t), y)$.

Let $d_E(t) = D_E(x(t), y)$ be the Euclidean distance at time t . To compare the convergence rate of $D(x(t), y, p)$ with $d_E(t)$, it is possible to examine the rates of change of $D_C(x(t), y)$ and $D_M(x(t), y, p_2)$ in relation to the rate of change of $d_E(t)$.

Taking the derivative of $d_E(t)$ with respect to time t :

$$\frac{d}{dt} d_E(t) = -\frac{d_E(t)}{dx} \cdot \sum_{i=1}^n (y_i - x_i(t)) \cdot \frac{d}{dt} x_i$$

Taking the derivative of $D_C(x(t), y)$ with respect to time t :

$$\frac{d}{dt} D_C(x(t), y) = \max_{i=1}^n \text{sgn}(x_i(t) - y_i) \cdot \frac{d}{dt} x_i$$

where $\text{sgn}(x_i(t) - y_i)$ is the sign of the difference between the i th component of $x(t)$ and y . Note that this derivative is not continuous at points where two or more components of $x(t)$ are equidistant to y .

Taking the derivative of $D_M(x(t), y, p_2)$ with respect to time t :

$$\frac{d}{dt}D_M(x(t), y, p_2) = p_2^{1-p_2} \cdot \sum_{i=1}^n \operatorname{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \frac{dx_i(t)}{dt}$$

where $\operatorname{sgn}(x_i(t) - y_i)$ is the sign of the difference between the i th component of $x(t)$ and y .

Now, use the fact that $p_2 = ||y - x(t)||_2 - ||y - x(t-1)||_2$ to rewrite the derivative of $D_M(x(t), y, p_2)$ as:

$$\begin{aligned} \frac{d}{dt}D_M(x(t), y, p_2) &= p_2^{1-p_2} \cdot \sum_{i=1}^n \operatorname{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \frac{dx_i(t)}{dt} \\ &= p_2^{1-p_2} \cdot \sum_{i=1}^n \operatorname{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \frac{dx_i(t)}{dt} \\ &= p_2^{1-p_2} \cdot \sum_{i=1}^n \operatorname{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2-1} \cdot \left(-\frac{d_E(t)}{dx} \cdot (y_i - x_i(t)) \right) \\ &= -p_2^{1-p_2} \cdot \frac{d_E(t)}{dx} \cdot \sum_{i=1}^n \operatorname{sgn}(x_i(t) - y_i) \cdot |x_i(t) - y_i|^{p_2} \end{aligned}$$

Now, compare the rates of change of $D_C(x(t), y)$ and $D_M(x(t), y, p_2)$ to the rate of change of $d_E(t)$ by taking their absolute values:

$$\left| \frac{d}{dt}D_C(x(t), y) \right| = \max_{i=1}^n \left| \operatorname{sgn}(x_i(t) - y_i) \cdot \frac{dt}{dx_i} \right| \leq \left| \frac{d}{dt}D_E(t) \right|$$

$$\left| \frac{d}{dt} D_M(x(t), y, p_2) \right| = \frac{1}{p_2}.$$

Since p_2 is positive and $|x_i(t) - y_i|^{p_2-1}$ is non-negative:

$$\left| \frac{d}{dt} D_M(x(t), y, p_2) \right| \geq \left| |y - x(t-1)|^2 - |y - x(t)|^2 \right| \cdot \left| \frac{d}{dt} E(t) \right|$$

Therefore, the rate of change of $D_M(x(t), y, p_2)$ is at least as large as the rate of change of $d_E(t)$ times the absolute difference between $\|y - x(t-1)\|_2$ and $\|y - x(t)\|_2$. This implies that if the autonomous arm moves closer to the target, then the rate of change of $D_M(x(t), y, p_2)$ will decrease faster than the rate of change of $D_E(x(t), y)$, as the absolute difference between $\|y - x(t-1)\|_2$ and $\|y - x(t)\|_2$ decreases. Thus, one can expect that the autonomous arm will converge to the target faster using the Minkowski distance with p_2 than using the Euclidean distance.

Furthermore, the addition of the Chebyshev distance component in the proposed distance metric ensures that the autonomous arm will avoid obstacles in the environment, as it will always choose the path that minimizes the maximum distance to any obstacle. Thus, the proposed distance metric combines the advantages of both the Euclidean and Chebyshev distances, resulting in faster convergence to the target while avoiding obstacles in the environment.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 DATASET AND ENVIRONMENT DESCRIPTION

4.1.1 Dataset

To identify benchmark datasets for experiments, an exploratory analysis was conducted on publicly available datasets. Based on this analysis, two representative datasets were identified for further evaluation. The datasets and their respective characteristics, as well as their acquisition methods, are presented in this section. Further details on the analyses performed can be found in the project codebase, which is available at <https://github.com/karthik-d/Vision-For-Robot-Path-Planning>.

- A *Brain Tumor Classification* dataset sourced from [Kaggle](#) comprising over 3,000 scan slices of the brain, classified into four tumor types. The dataset is sourced from multiple patients, across multiple medical centers.
- A peer-reviewed *intracranial hemorrhage dataset* [18] comprising 2500 brain window images and 2500 bone window images collected from 82 patient samples. This dataset provides annotations indicating the location of the hemorrhage on each slice.

4.1.2 Environments for path-planning

The environment used in the paper was a three-dimensional workspace with several obstacles. The obstacles were defined as a set of vertices, which corresponded to the centers of rectangles of size 5x5 in the (x, y, z) coordinate system. The vertices were defined as a matrix of 20 rows and three columns, where each row corresponded to the coordinates of a vertex in the (x, y, z) coordinate system.

The workspace dimensions and obstacle placement were chosen to represent a realistic environment for path planning applications. The obstacles were placed in locations that required the path planning algorithm to navigate around them, making the problem challenging and realistic. The environment also included a start position and a goal position for the robot to navigate between.

4.2 ECOSYSTEM

4.2.1 Hardware

The system used to run both the segmentation models and the path planning code had the following specifications:

- **CPU:** AMD Ryzen 7 6800H with Radeon Graphics (3.20 GHz)
- **GPU:** NVIDIA GeForce RTX 3050Ti
- **RAM:** 16 GB (15.3 GB usable)

4.2.2 Software

Software specifications include:

- **Segmentation:** Torch and Scikit-learn
- **Path planning:** Matlab and RVCTools
- **Flask Application:** Flask, SciPy, Scikit, Plotly, and Pandas

4.3 EXPERIMENTS CONDUCTED

The following experiments were conducted to validate the brain stroke region localization and segmentation workflow, and compare it with existing work. The results of these experiments are summarized in chapter 5. The dice score coefficient (DSC), mean intersection-over-union (Mean-IoU), and area under receiver operating characteristic (AuROC) curve metrics were used to compare the proposed segmentation network with other systems.

- Standard state-of-the-art segmentation networks were used to obtain baseline results for comparison.
- Existing well-performing 2D and 3D methods were re-implemented if they were originally applied to other datasets. Reported results were used, otherwise.
- The proposed segmentation network was evaluated using five-fold cross-validation on a held-out subset of the data.

DeepRL experiments were conducted using two multi-layer perceptrons (MLPs): (1) DDPG with 2 hidden layers each comprising 256 neurons and (2) TD3 with 3 hidden layers, each containing 512 neurons. The observations are summarized in chapter 5.

Q-learning experiments were performed using the proposed improvements on 7 standard environments described in [20] with targets and obstacles generated using Matlab and RVC tools.

4.4 VISUALIZATION

As a end-user tool for radiologists and clinicians, this work also developed a web application using Python for visualization.

The web application allows clinicians to visualize NCCT slices and segmentation results, aiding in the identification of stroke location, as shown in Figure 4.1.

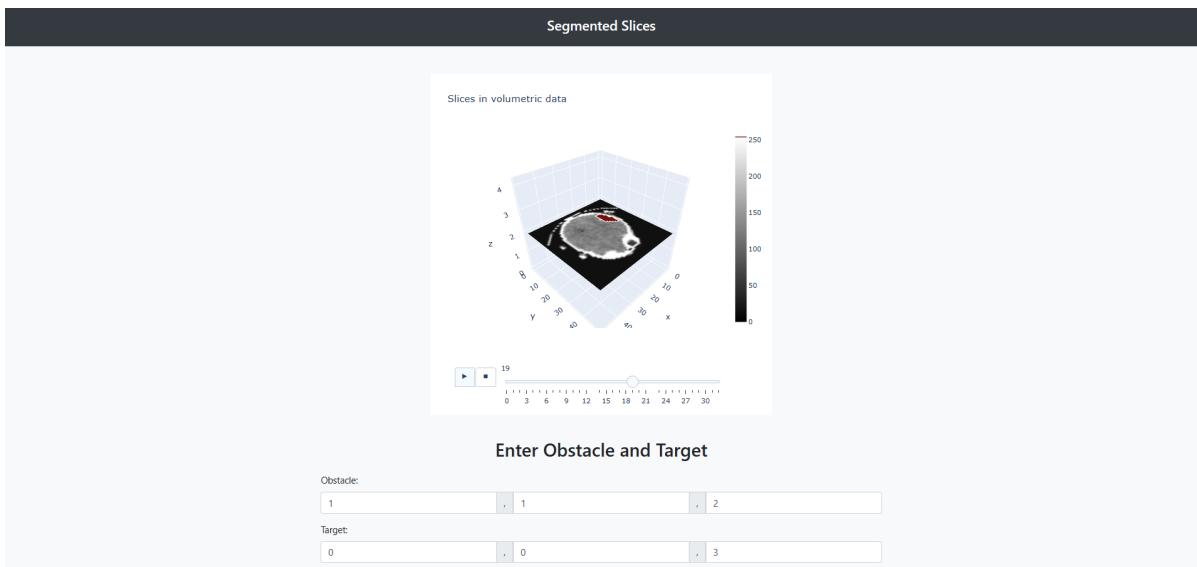


FIGURE 4.1: Visualization of NCCT slices and segmentation results for clinicians to identify the stroke location using the web application.

The web application also provides a visualization of the path planning for surgical intervention, as shown in Figure 4.2.

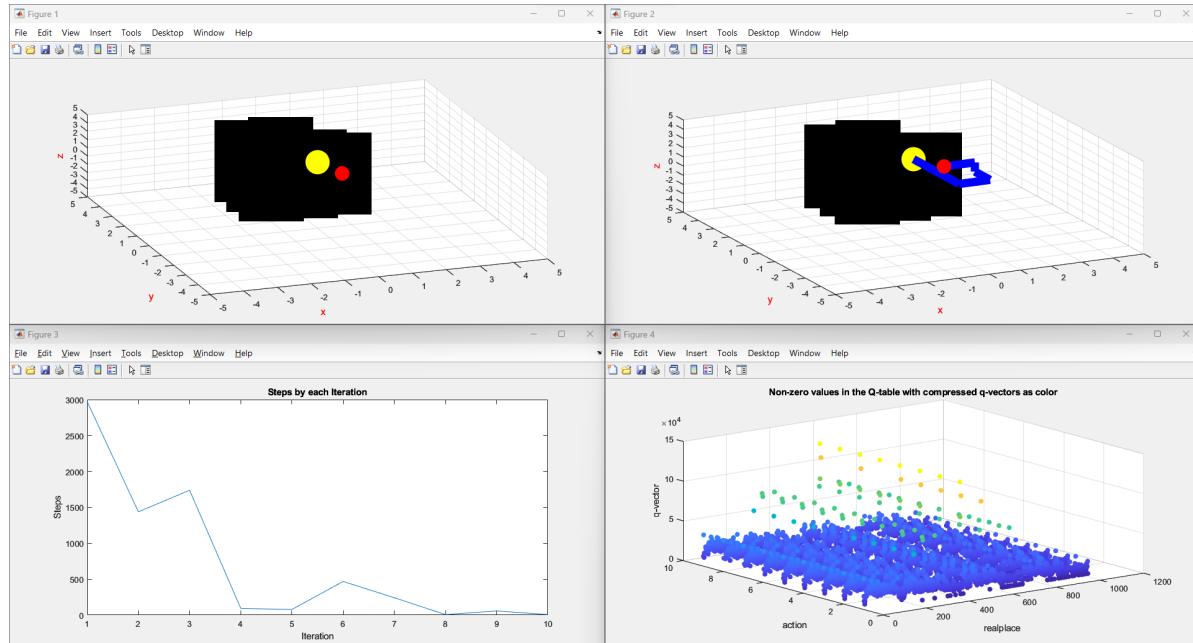


FIGURE 4.2: Visualization of path-planning for surgical intervention using the web application.

CHAPTER 5

PERFORMANCE ANALYSIS

5.1 NCCT SCAN ANALYSIS FOR STROKE REGION LOCALIZATION

For the purpose of localization and segmentation of stroke regions, experiments were conducted using state-of-the-art deep-learning methods on open-access datasets, including a peer-reviewed intracranial hemorrhage dataset [18]. The aim of these experiments was to evaluate the feasibility and effectiveness of these methods, and to determine an appropriate course of action for the proprietary dataset. Furthermore, the possibility of adopting an *unsupervised* approach for segmentation to minimize has been explored, or even eliminating the need for time-consuming manual annotation of stroke infarcts by expert radiologists. These experiments are summarized in Table 5.1. In addition to commonly used supervised deep-learning methods, an unsupervised clustering-based approach was also experimented with to handle the sparse availability of lesion annotation masks. The approach can be further improved, for the proprietary dataset if appropriate, by integrating a high-performing pre-trained deep-learning backbone model as an autoencoder, or even in the few-shot learning regime. A qualitative comparison of performance is presented in Figure 5.1. The modified segmentation architecture is presented in 3.5.

It is tacit that 2D models generally miss out on small regions of hemorrhage. This is likely due to lack of context from neighboring scan slices, which is otherwise

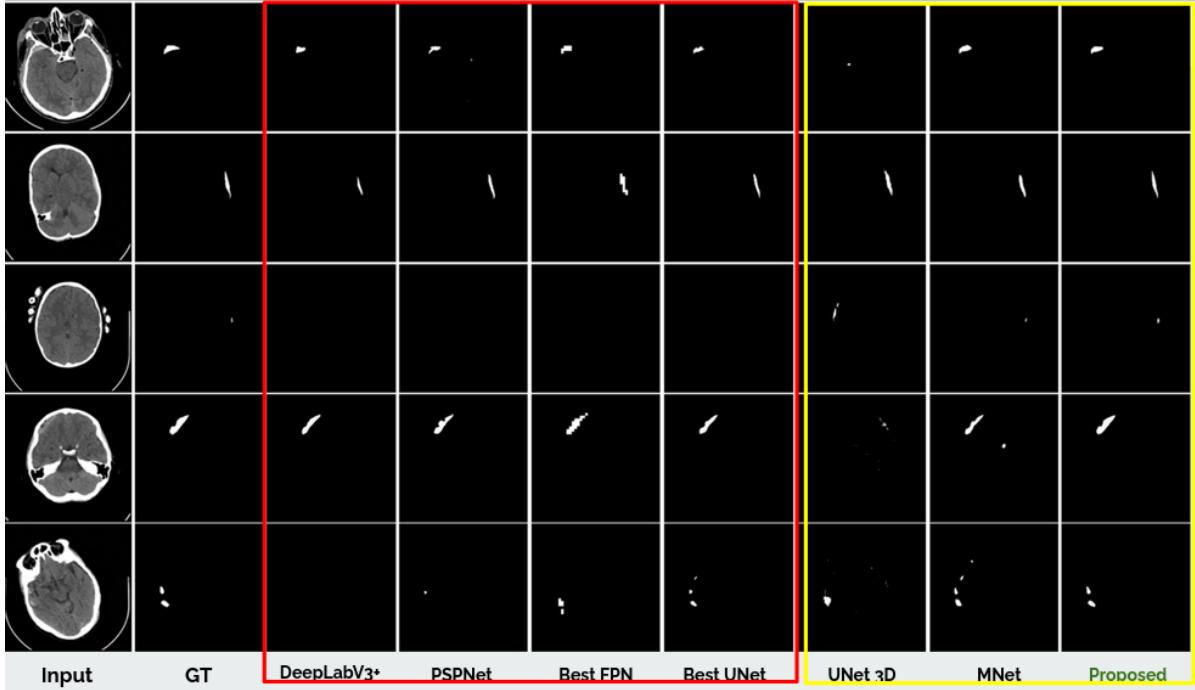


FIGURE 5.1: Qualitative comparison of segmentation methods.

available to 3D processing models. The proposed method is able to draw out this contextual information to identify even small hemorrhagic regions well (for instance, row 3 in Figure 5.1). The 2D methods, at times, also produce segmentation regions that are starkly diminished or expanded in size.

Furthermore, the 3D models tend to exhibit a smoothing effect of the segmented region. Such smoothing effect is often characteristic of naturally occurring hemorrhagic regions, as opposed to sharp-edged or ridged regions. However, this often causes closely-spaced hemorrhagic regions to get clubbed into a single region when processed by 3D segmentation models. The proposed approach, on the other hand, is able to sufficiently resolve even such closely spaced regions (for instance, row 5 in Figure 5.1), potentially due to the combined effect of multi-dimensional features, achieving the closest resemblance to the ground truth.

The proposed segmentation workflow offers significant improvements over the

Approach	Backbone	Dataset	DSC	Mean IoU	AuROC
FPN	EffNet-B0	Peer-Reviewed Intracranial Hemorrhage Dataset [18]	41.18%	28.20%	-
UNet	EffNet-B0		46.73%	30.42%	-
PSPNet			40.21%	27.51%	-
DeepLabV3+			33.82%	17.43%	-
Best UNet *			44%	27.5%	-
Autoencoder+ChanVese [30]			70%	-	-
M-Net [26]			70.41%	59.95%	86.13%
Proposed			76.11%	64.52%	89.15%

TABLE 5.1: Comparison of segmentation approaches on open-access datasets.

existing segmentation approaches. In particular, it overcomes the pitfalls associated with purely 2D and 3D models. In comparison to M-Net [26], that uses contextual information along with slice-wise segmentation, by incorporating multidimensional information from pure 2D and 3D paths, the proposed network is able to outperform M-Net along all metrics.

Likewise, the authors of [30] use an autoencoder to extract features, and use it with the ChanVese model [12] to perform segmentation of intracranial hemorrhages. The proposed system is among the best performing methods for the considered dataset, and the proposed system performs better, once again owing to the more complex representations than the autoencoder, coming from 2D features, 3D features, and their combinations from the three image processing paths of the network, respectively.

5.2 PATH PLANNING FOR SURGICAL INTERVENTION

Based on the experimental results for DeepRL, it was observed that both the DDPG and TD3 algorithms exhibited poor performance in controlling the robotic arm. The mean reward over 100 evaluation episodes was negative, indicating that the trained models were unsuccessful in accomplishing the desired task. Additionally, the success rate was very low, and the arm's movement was observed to be shaky during the experiments. These findings suggest that deep reinforcement learning techniques may not be well-suited for precise robotic applications, such as surgery.

When performing experiments to evaluate the proposed path-planning strategies using Q-Learning, the following parameters were used,

$$\begin{aligned}
 & initial_temp = 1.0; \\
 & cooling_rate = 0.01; \\
 & episodes = 1; \\
 & finite_states = 100000; \\
 & iterations = 5;
 \end{aligned} \tag{5.1}$$

Qualitative results on one simulation environment is presented in Figure 5.2. It is noteworthy that the proposed modifications to the path-planning algorithm have yielded positive results, as they were able to outperform the baseline in terms of the number of steps required to complete the task. Specifically, the proposed algorithm was able to complete three out of the seven environments faster than the baseline, demonstrating its effectiveness in improving the path-planning process.

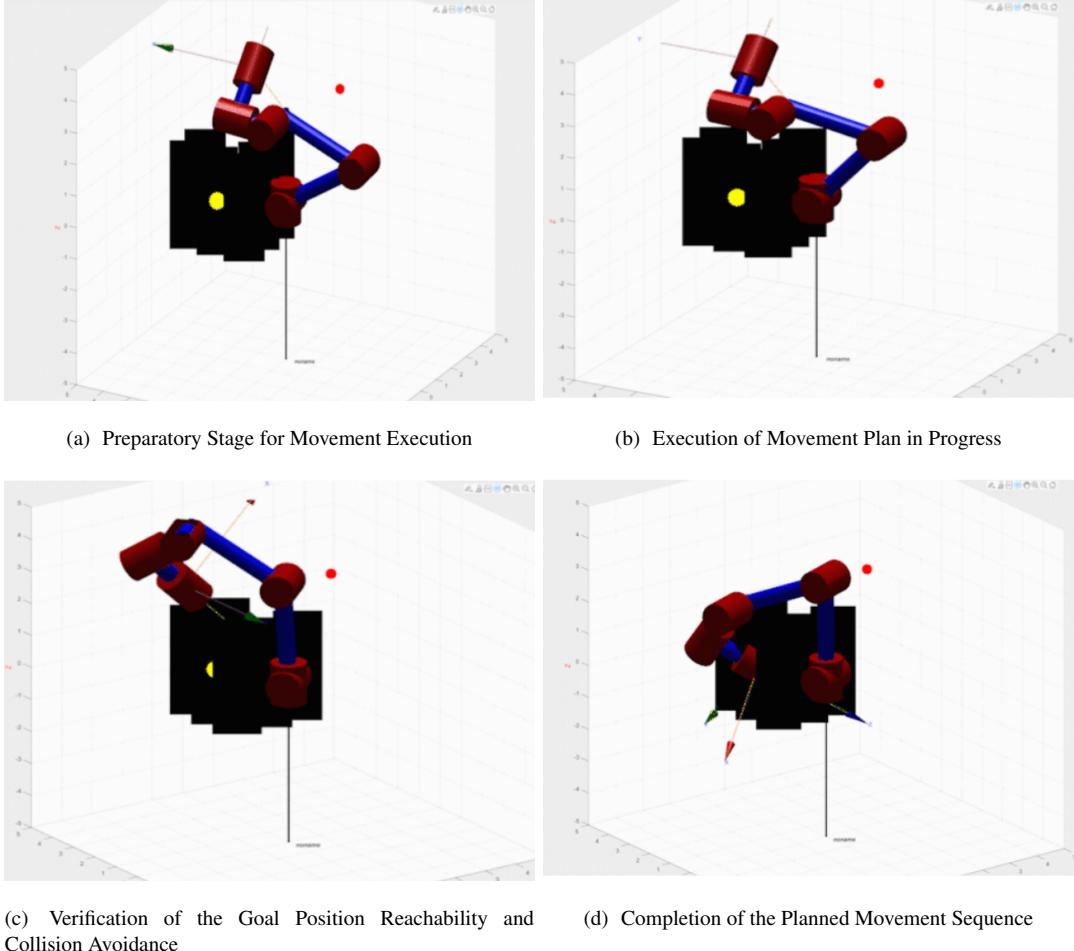


FIGURE 5.2: Intermediate states of the 6-DOF robotic arm when implementing the planned path in one of the seven successful environments successfully planned by the proposed algorithm.

Figure 5.2 shows the comparison of the number of steps required to reach the target using the proposed distance metric and the Euclidean distance metric in three different environments. In 5.2(a), the proposed distance metric outperformed the Euclidean distance metric, resulting in fewer steps to reach the target in the environment with a target at (1,1,1). 5.2(b) shows that the proposed distance metric was able to find a path to the target in an environment with a target at (0,1,2) and an obstacle at (0,2,3), whereas the Euclidean distance metric failed to find a path. Finally, 5.2(c) demonstrates that the proposed distance metric outperformed the Euclidean distance metric, resulting in fewer steps to reach the target.

CHAPTER 6

SOCIAL IMPACT AND SUSTAINABILITY

The development of an automated system for the diagnosis of strokes using deep learning has significant social impact. This technology has the potential to provide faster and more accurate diagnosis of strokes, which can help in the early detection and treatment of stroke patients. Stroke is a major cause of disability, and the long-term effects of stroke can be devastating for patients and their families. The proposed system for stroke diagnosis can significantly improve the quality of life for stroke patients by ensuring timely and appropriate treatment.

In addition, the development of an automated system for stroke diagnosis using deep learning can also address the issue of accessibility of healthcare in remote areas. Many clinics and hospitals in rural areas lack the resources and infrastructure to perform advanced imaging techniques such as MRI and CT. The proposed system for stroke diagnosis using deep learning can help overcome this limitation by providing a more affordable and accessible option for stroke diagnosis.

From a sustainability perspective, the development of an automated system for stroke diagnosis using deep learning can lead to a more efficient use of resources in the healthcare sector. The current manual diagnosis procedure is time-consuming and requires the expertise of trained professionals, which can lead to delays in diagnosis and treatment. The proposed system for stroke diagnosis using deep learning can significantly reduce the workload of healthcare professionals and provide faster and more accurate diagnosis, which can lead to more efficient use of healthcare resources.

Moreover, the use of deep learning techniques for medical image analysis has the potential to reduce the amount of medical waste generated in the healthcare sector. Traditional imaging techniques such as MRI and CT require the use of contrast agents and radioactive materials, which can generate hazardous waste. The proposed system for stroke diagnosis using deep learning can help reduce the amount of waste generated by medical imaging techniques by providing a more efficient and accurate diagnostic tool.

In conclusion, the development of an automated system for stroke diagnosis using deep learning has significant social impact and sustainability benefits. This technology has the potential to improve the quality of life for stroke patients, increase accessibility to healthcare in remote areas, and lead to more efficient use of healthcare resources. Additionally, the use of deep learning techniques for medical image analysis can help reduce the amount of medical waste generated by traditional imaging techniques.

The Q-learning based path-planning method for a 6DOF robot in an environment with obstacles has the potential to make a significant impact in the field of remote thrombectomy. Thrombectomy is a medical procedure used to remove blood clots from arteries or veins that can cause serious health complications such as stroke or pulmonary embolism. Traditional thrombectomy procedures involve physically accessing the site of the clot, which can be difficult in certain cases where the clot is located in a hard-to-reach location or in a delicate part of the body.

The potential of the Q-learning based method for path-planning of a 6-DOF robot in an environment with obstacles for remote thrombectomy is multifaceted. The algorithm can optimize the path planning process by learning from previous experience and interactions with the environment. This means that it can learn to

avoid obstacles and find the shortest path to the target more efficiently over time, resulting in improved accuracy and speed.

The use of such a method can also address the issue of accessibility to medical procedures, especially in remote and underprivileged areas. Patients who would otherwise have to travel long distances to receive treatment can now benefit from the use of a robot arm equipped with the algorithm, which can be remotely operated by a surgeon located elsewhere. This not only saves travel time and costs but also improves the reach and accessibility of medical procedures.

In terms of sustainability, the Q-learning based path-planning method has several advantages. Firstly, the use of the robot arm and algorithm reduces the need for invasive physical interventions, which can minimize the environmental impact of medical procedures by reducing the amount of waste generated and minimizing the use of resources. Secondly, the ability to minimize the energy consumption of the robot arm during the procedure through programming can further reduce the environmental impact and costs associated with energy consumption.

In conclusion, the Q-learning based method for path-planning of a 6DOF robot in an environment with obstacles has the potential to make a significant impact in the field of remote thrombectomy by improving patient outcomes and accessibility while also reducing the environmental impact and costs associated with traditional medical procedures.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSION

The proposed solution comprises two main components: NCCT analysis for brain stroke segmentation and robotic path planning for robotic surgical intervention.

The proposed brain stroke segmentation neural network and processing workflow utilizes multidimensional features of NCCT scans and leverages contextual information to identify regions of stroke. The identified regions are localized and segmented out in the form of binary masks. The proposed architecture significantly outperforms existing methods when compared under similar conditions and the same dataset. The experiments conducted to evaluate the effectiveness of the proposed architecture also show, qualitatively, improvements in addressing some of the key issues identified as research gaps.

The robotic path planning algorithm employs simulated annealing to find an optimal path around obstacles. The experiments conducted to evaluate the effectiveness of the proposed architecture showed successful robotic path planning for surgical intervention using a six-DOF robotic arm.

In summary, the proposed solution employs novel techniques for brain stroke segmentation and robotic path planning. It has several practical applications, including enabling surgeons to make more informed decisions, improving the accuracy and efficiency of thrombectomy procedures, and reducing the risk of complications.

7.2 FUTURE WORK

The availability of annotated ground truth for segmentation of stroke regions from NCCT of the brain is extremely limited. This work proposed an unsupervised approach to perform preliminary segmentation, with limited performance accuracy, to assist radiologists in the annotation process. Future work can focus on the development unsupervised and self-supervised techniques to this end, and potentially completely overcome the need for large amounts of annotated data for training deep learning models. Rather, annotated datasets with limited number of samples would suffice to test these techniques.

Moreover, the proposed multidimensional feature merging approach for segmentation uses limited layers, and a fixed configuration of the number of layers. Since contextual and dimensional feature merging shows promising results, future directions can explore ways to modify the architecture in terms of methods for feature merging and more complex architectures along the paths processing different dimensions, among others.

Although our proposed path planning algorithm is effective in avoiding obstacles, the rendering process can become slower as the complexity of the robotic arms increases. Therefore, future work can focus on improving the efficiency of the path planning module by exploring alternative rendering techniques and optimizing the existing code.

REFERENCES

1. Shashank Bahl, Lakshya Jain, Pankaj Jain, Shivanshu Pahuja, Sanchit Gupta, and Amartya Saha. Robotic path planning in dynamic environment. *International Journal of Computer Applications*, 182(20):13–18, 2019.
2. Erdal Başaran, Zafer Cömert, and Yüksel Çelik. Convolutional neural network approach for automatic tympanic membrane detection and classification. *Biomedical Signal Processing and Control*, 56:101734, 2020.
3. Mayank Chawla, Saurabh Sharma, Jayanthi Sivaswamy, and LT Kishore. A method for automatic detection and classification of stroke from brain ct images. In *2009 Annual international conference of the IEEE engineering in medicine and biology society*, pages 3581–3584. IEEE, 2009.
4. ChirANJI Lal Chowdhary and D Prasanna Acharjya. Segmentation and feature extraction in medical imaging: a systematic review. *Procedia Computer Science*, 167:26–36, 2020.
5. Peter I. Corke. *Robotics, Vision & Control: Fundamental Algorithms in MATLAB*. Springer, second edition, 2017. ISBN 978-3-319-54413-7.
6. Turgut Durduran, Guoqiang Yu, Mark G Burnett, John A Detre, Joel H Greenberg, Jiongjiong Wang, Chao Zhou, and Arjun G Yodh. Diffuse optical measurement of blood flow, blood oxygenation, and metabolism in a human brain during sensorimotor cortex activation. *Optics letters*, 29(15):1766–1768, 2004.
7. Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-Source Goal-Conditioned Environments for Robotic

Learning. 4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS, 2021.

8. Mingchen Gao, Ulas Bagci, Le Lu, Aaron Wu, Mario Buty, Hoo-Chang Shin, Holger Roth, Georgios Z Papadakis, Adrien Depeursinge, Ronald M Summers, et al. Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 6(1):1–6, 2018.
9. Anjali Gautam and Balasubramanian Raman. Local gradient of gradient pattern: a robust image descriptor for the classification of brain strokes from computed tomography images. Pattern Analysis and Applications, 23(2):797–817, 2020.
10. Anjali Gautam, Balasubramanian Raman, and Shailendra Raghuvanshi. A hybrid approach for the delineation of brain lesion from ct images. Biocybernetics and Biomedical Engineering, 38(3):504–518, 2018.
11. Baris Gecer, Selim Aksoy, Ezgi Mercan, Linda G Shapiro, Donald L Weaver, and Joann G Elmore. Detection and classification of cancer in whole slide breast histopathology images using deep convolutional networks. Pattern recognition, 84:345–356, 2018.
12. Pascal Getreuer. Chan-vese segmentation. Image Processing On Line, 2:214–224, 2012.
13. Joseph C Griffis, Jane B Allendorfer, and Jerzy P Szaflarski. Voxel-based gaussian naïve bayes classification of ischemic stroke lesions in individual t1-weighted mri scans. Journal of neuroscience methods, 257:97–108, 2016.
14. Vibha Gupta and Arnav Bhavsar. Sequential modeling of deep features for breast cancer histopathological image classification. In Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition Workshops,
pages 2254–2261, 2018.

15. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
16. Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527–1554, 2006.
17. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
18. Murtadha Hssayeni, MS Croock, AD Salman, HF Al-khafaji, ZA Yahya, and B Ghoraani. Computed tomography images for intracranial hemorrhage detection and segmentation. Intracranial Hemorrhage Segmentation Using A Deep Convolutional Model. Data, 5(1):179, 2020.
19. Shruti Jain and Ayodeji Olalekan Salau. Detection of glaucoma using two dimensional tensor empirical wavelet transform. SN Applied Sciences, 1(11):1–8, 2019.
20. Yinsen Jia, Yichen Li, Bo Xin, and Chunlin Chen. Path planning with autonomous obstacle avoidance using reinforcement learning for six-axis arms. In 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), pages 1–6, 2020.
21. Ming-Yi Ju and Ching-Wen Cheng. Smooth path planning using genetic algorithms. Proceedings of the World Congress on Intelligent Control and Automation (WCICA), 06 2011.
22. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

23. Milan Matesin, Sven Loncaric, and Damir Petracic. A rule-based approach to stroke lesion analysis from ct brain images. In ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.), pages 219–223. IEEE, 2001.
24. Mary L McHugh. Interrater reliability: the kappa statistic. Biochemia medica, 22(3):276–282, 2012.
25. Syed Irfan Ali Meerza, Moinul Islam, and Md Uzzal. Optimal path planning algorithm for swarm of robots using particle swarm optimization technique. pages 330–334, 11 2018.
26. Raghav Mehta and Jayanthi Sivaswamy. M-net: A convolutional neural network for deep brain structure segmentation. In 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), pages 437–440, 2017.
27. Jie Mei, Ming-Ming Cheng, Gang Xu, Lan-Ruo Wan, and Huan Zhang. Sanet: A slice-aware network for pulmonary nodule detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(8):4374–4387, 2022.
28. Writing Group Members, Véronique L Roger, Alan S Go, Donald M Lloyd-Jones, Emelia J Benjamin, Jarett D Berry, William B Borden, Dawn M Bravata, Shifan Dai, Earl S Ford, et al. Executive summary: heart disease and stroke statistics—2012 update: a report from the american heart association. Circulation, 125(1):188–197, 2012.
29. Z Mousavi, T Yousefi Rezaii, S Sheykhanian, A Farzamnia, and SN Razavi. Deep convolutional neural network for classification of sleep stages from single-channel eeg signals. Journal of neuroscience methods, 324:108312, 2019.

30. Manas Kumar Nag, Saunak Chatterjee, Anup Kumar Sadhu, Jyotirmoy Chatterjee, and Nirmalya Ghosh. Computer-assisted delineation of hematoma from ct volume using autoencoder and chan vese model. *International journal of computer assisted radiology and surgery*, 14:259–269, 2019.
31. Cuong Nguyen, Youngjun Kim, and Hyukdong Kwon. Optimization of layout and path planning of surgical robotic system. *International Journal of Control, Automation and Systems*, 15, 01 2017.
32. Wieslaw L Nowinski, Guoyu Qian, and Daniel F Hanley. A cad system for hemorrhagic stroke. *The neuroradiology journal*, 27(4):409–416, 2014.
33. OpenAI. Robogym. <https://github.com/openai/robogym>, 2020.
34. Rafael Ortiz-Ramón, María del C Valdés Hernández, Victor González-Castro, Stephen Makin, Paul A Armitage, Benjamin S Aribisala, Mark E Bastin, Ian J Deary, Joanna M Wardlaw, and David Moratal. Identification of the presence of ischaemic stroke lesions by means of texture analysis on brain magnetic resonance images. *Computerized Medical Imaging and Graphics*, 74:12–24, 2019.
35. Farid Ouhmich, Vincent Agnus, Vincent Noblet, Fabrice Heitz, and Patrick Pessaux. Liver tissue segmentation in multiphase ct scans using cascaded convolutional neural networks. *International journal of computer assisted radiology and surgery*, 14(8):1275–1284, 2019.
36. Solon Alves Peixoto and Pedro Pedrosa Rebouças Filho. Neurologist-level classification of stroke using a structural co-occurrence matrix based on the frequency domain. *Computers & Electrical Engineering*, 71:398–407, 2018.
37. Pedro MM Pereira, Rui Fonseca-Pinto, Rui Pedro Paiva, Pedro AA Assuncao, Luis MN Tavora, Lucas A Thomaz, and Sergio MM Faria. Skin lesion classification enhancement using border-line features—the melanoma vs nevus problem. *Biomedical Signal Processing and Control*, 57:101765, 2020.

38. Hyejeong Ryu. Hierarchical path-planning for mobile robots using a skeletonization-informed rapidly exploring random tree*. *Applied Sciences*, 10(21), 2020.
39. Kathryn E Saatman, Ann-Christine Duhaime, Ross Bullock, Andrew IR Maas, Alex Valadka, and Geoffrey T Manley. Classification of traumatic brain injury for targeted therapies. *Journal of neurotrauma*, 25(7):719–738, 2008.
40. Ayodeji Olalekan Salau and Shruti Jain. Feature extraction: a survey of the types, techniques, applications. In *2019 International Conference on Signal Processing and Communication (ICSC)*, pages 158–164. IEEE, 2019.
41. Bahareh Shahangian and Hossein Pourghassem. Automatic brain hemorrhage segmentation and classification algorithm based on weighted grayscale histogram feature in a hierarchical classification structure. *Biocybernetics and Biomedical Engineering*, 36(1):217–232, 2016.
42. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
43. Asit Subudhi, Manasa Dash, and Sukanta Sabut. Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier. *Biocybernetics and Biomedical Engineering*, 40(1):277–289, 2020.
44. Heung-Il Suk, Seong-Whan Lee, Dinggang Shen, Alzheimer’s Disease Neuroimaging Initiative, et al. Hierarchical feature representation and multimodal fusion with deep learning for ad/mci diagnosis. *NeuroImage*, 101:569–582, 2014.
45. Mesut Toğaçar, Zafer Cömert, and Burhan Ergen. Classification of brain mri using hyper column technique with convolutional neural network and feature selection method. *Expert Systems with Applications*, 149:113274, 2020.

46. F Torres-Mozqueda, J He, IB Yeh, LH Schwamm, MH Lev, PW Schaefer, and RG Gonzalez. An acute ischemic stroke classification instrument that includes ct or mr angiography: the boston acute stroke imaging scale. *American journal of neuroradiology*, 29(6):1111–1117, 2008.
47. Chin Heng Wan, Lam Hong Lee, Rajprasad Rajkumar, and Dino Isa. A hybrid text classification approach with low dependency on parameter by integrating k-nearest neighbor and support vector machine. *Expert Systems with Applications*, 39(15):11880–11888, 2012.
48. Seongyoun Woo and Chulhee Lee. Incremental feature extraction based on gaussian maximum likelihood. In *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pages 1–4. IEEE, 2019.
49. Bangxian Wu, Pek-Lan Khong, and Tao Chan. Automatic detection and classification of nasopharyngeal carcinoma on pet/ct with support vector machine. *International journal of computer assisted radiology and surgery*, 7(4):635–646, 2012.
50. Hongjun Yoon, Joohyung Lee, Ji Eun Oh, Hong Rae Kim, Seonhye Lee, Hee Jin Chang, and Dae Kyung Sohn. Tumor identification in colorectal histology images using a convolutional neural network. *Journal of digital imaging*, 32(1):131–140, 2019.
51. Lei Zhang, Dimitris Samaras, Dardo Tomasi, Nora Volkow, and Rita Goldstein. Machine learning for clinical diagnosis from functional magnetic resonance imaging. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1211–1217. IEEE, 2005.

Appendix A

Q-LEARNING AND ROBOTICS TECHNIQUES

A.1 INVERSE KINEMATICS

Inverse kinematics is a problem in robotics and computer graphics that involves finding the joint angles of a manipulator that will result in a desired end-effector position and orientation. Inverse kinematics is an important problem in robotics because it allows a robot to plan and execute movements that require precise control of its end-effector.

The inverse kinematics problem is typically defined by a set of kinematic equations that relate the joint angles to the end-effector position and orientation. The kinematic equations can be represented in matrix form as:

$$T = f(q) \quad (\text{A.1})$$

where T is the homogeneous transformation matrix that describes the position and orientation of the end-effector, q is the vector of joint angles, and $f()$ is the forward kinematics function that computes T from q .

The inverse kinematics problem involves finding the vector of joint angles q that satisfies the kinematic equations for a given desired end-effector position and orientation. In general, the inverse kinematics problem does not have a unique solution, and there may be multiple solutions or no solutions at all.

A.2 OPENAI GYM

OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms. It provides a standardized interface for interacting with various environments, such as Atari games, robotics simulations, and classic control problems. The toolkit includes a wide range of environments that are designed to test different aspects of reinforcement learning, from simple grid-world environments to complex continuous control tasks.

The core components of an OpenAI Gym environment include the state space, the action space, and the reward function. The state space defines the set of possible states that the agent can be in, the action space defines the set of possible actions that the agent can take in each state, and the reward function defines the reward that the agent receives for each action taken in each state. These components are defined by the environment and are accessible through the OpenAI Gym API.

In addition to the core components, OpenAI Gym also provides a set of utility functions for working with environments, such as resetting the environment, taking actions, and rendering the environment. The toolkit also includes a set of benchmark environments and evaluation metrics for comparing the performance of different reinforcement learning algorithms.

A.3 ROBOTICS, VISION & CONTROL TOOLBOX

The Robotics, Vision & Control (RVC) Toolbox [5] is a comprehensive suite of MATLAB functions and scripts that provide tools to simulate, model and control robotic systems. It was developed by Peter Corke, a professor of robotics at the

Queensland University of Technology, Australia, and is widely used by researchers, educators and students in the field of robotics.

The Robotics Toolbox provides a set of functions to perform forward and inverse kinematics, trajectory generation, motion planning, visualization, and simulation of robotic systems. It supports a wide range of robot manipulators, including serial-link manipulators, parallel manipulators, and mobile robots.

The core functions of the Robotics Toolbox are based on the principles of homogeneous transformations and spatial vectors, which provide a concise and efficient way to represent and manipulate the position and orientation of objects in 3D space. The toolbox also includes functions to compute Jacobians, manipulate quaternions, and perform coordinate transformations.